

UM SISTEMA PERICIAL PARA O ORDENAMENTO
BIOFISICO DO TERRITÓRIO
INTERROGÁVEL EM PORTUGUÊS - ORBI
RELATÓRIO FINAL - ANO I

LUIS MONIZ PEREIRA
EUGÉNIO OLIVEIRA
PAUL SABATIER

NÚCLEO DE INTELIGÊNCIA ARTIFICIAL
FCT/DI - 19/82
OUTUBRO 82

INTRODUÇÃO

=====

O presente relatório final diz respeito ao trabalho realizado, sob contracto entre o núcleo de Inteligência Artificial do Departamento de Informática da Universidade Nova de Lisboa e o Serviço de Estudos do Ambiente da Secretaria de Estado do Ambiente, em 1981/82, com vista à realização de um protótipo de um sistema informático para a avaliação dos recursos biofísicos do território com vista ao seu ordenamento.

Um segundo contracto, já em curso, instalara' o sistema em equipamento próprio do Serviço e reformula'-lo-a' de acordo com a experiência da sua utilização.

O presente relatório final, relativo ao ano I do projecto em curso, diz respeito nomeadamente a um sistema informático, interrogável num subconjunto do Português, e implementado na linguagem Prolog num pequeno computador, o qual para isso incorpora utiliza e assimila conhecimento de especialistas no domínio do ordenamento biofísico do território. Tal sistema é além disso capaz de explicar a aplicação desse conhecimento a uma base de dados territoriais (digitalizados com o sistema) assim como responder a perguntas sobre as suas possibilidades linguísticas e estruturação do seu conhecimento facilitando assim a sua utilização. Certas respostas podem ser desenhadas automaticamente sobre um mapa, por meio de um traçador gráfico acoplado ao sistema.

Neste relatório, apresentam-se muitos exemplos de utilização real do sistema relativamente a um conjunto de cerca de 600 pontos no concelho de Sesimbra, bem como exemplos de aquisição de novo conhecimento especializado e de interrogação do sistema sobre os seus próprios conhecimentos.

Apresenta-se ainda uma listagem completa de todos os programas desenvolvidos para o efeito bem como da base de dados utilizada na exemplificação.

A descrição técnica informática para além das listagens comentadas aqui incluídas, já foi objecto de anteriores relatórios [1,2,3,4] oportunamente entregues. O conhecimento especializado relativo às regras de avaliação dos recursos biofísicos do território, bem como a base de dados de pontos digitalizados relativos a parte do concelho de Sesimbra, foi-nos fornecido

pelo Serviço de Estudos do Ambiente [5].

OBJECTIVOS ALCANÇADOS:

=====

Foram alcançados todos os objectivos propostos, nomeadamente:

- ~~leitura dos mapas por contornos usando mesa digitalizadora~~
- aceitar, quer para os factos quer para as regras, um parametro de medida da sua credibilidade e fornecer as conclusões uma credibilidade que deriva dos factos e das regras que a suportam
- poder ser interrogado a todo o momento com perguntas pontuais
- poder produzir conclusões mapeificadas para toda uma região
- aceitar a modificação expedita de factos e regras pelo proprio utilizador
- responder sobre o seu modo de utilização
- explicar as suas respostas
- dialogar em português versatil.

Alem disso, nesta ocasião o sistema ja foi instalado no nosso equipamento igual ao que vai ser adquirido pelo Serviço (como previsto no segundo contrato), com excepção de que o nosso equipamento só em principios de Janeiro virá a compreender um disco duro de 36,4 MB igual ao previsto para o Serviço, bem como a memoria de 256 KB também prevista. Em consequência da falta de espaço no disco flexível que utilizamos, o sistema não pode ainda lidar com uma base de dados de muitos milhares de pontos, e em consequência da limitação de memoria (estamos por enquanto reduzidos a 128 KB) os modulos (necessarios) de interrogacao sobre as capacidades linguisticas do sistema e de interrogacao por menu não cabem simultaneamente com o conjunto dos outros modulos.

Assim estes dois modulos, previstos para o prototipo do primeiro contrato, embora prontos e testados (apresentamos deles uma listagem nos anexos) não podem ser vistos a funcionar integrados no sistema completo dada a exiguidade de memoria disponível. Este contratempo, a remediar em Janeiro, deveu-se a atrasos na aquisição do material, que não dependeram de nos.

DOMINIO DA APLICACAO

=====

O dominio de conhecimento envolve a avaliacao da oferta biologica e fisica

do meio. A dimensao tipica da residuo e' a do concelho, a qual e' descrita por intermedio de 23 descriptors. A cada descriptor corresponde um mapa da residuo coberto por uma grelha uniforme, formando quadrados de 200 metros de lado. Em cada quadrado os valores dos descriptors sao digitalizados em inteiros de 1 a 5. Alem disso a cada valor e' associada uma representatividade, tambem um inteiro de 1 a 5, descrevendo o grau de homogeneidade do valor no quadrado considerado.

Cada concelho compreende em media 6000 quadrados. Exemplos de descriptors

d1 -- qualidade dos microclimas

d9 -- riscos de erosao

d11-- equilibrio da vegetacao

Dada a base de dados dos factos observados expressos pelos descriptors, informacao derivada mais abstracta pode ser obtida definindo factores e aptidoes. Um factor e' o resultado de uma combinacao de descriptors. Tem um valor inteiro entre 1 e 5 ou um valor s, significando ser esse factor selectivo. Tambem aqui a cada valor e' associada uma representatividade entre 1 e 5. O valor de cada factor e' computado atraves dos valores dos descriptors que o suportam atraves de uma regra fornecida pelos especialistas. Uma regra pode ter varios casos alternativos dependendo dos valores actuais dos descriptors. Por exemplo, um descriptor pode ter um valor que se sobrepe a todos os outros e como tal definir por si so' completamente o factor, ou ainda o valor do factor pode ser uma soma pesada dos valores de todos, ou de alguns, descriptors em que se baseia de acordo com os valores presentes. A representatividade dos factores contudo, e' sempre uma soma ponderada das representatividades dos descriptors intervenientes. Ha' 16 factores. Exemplos sao:

f1 -- conforto climatico

f3 -- resistencia do solo a erosao

f6b-- riquezas cinegeticas e paisagisticas

Aptidoes sao analogamente definidas em termos de factores, tal como estes em termos de descriptors, atraves de regras de inferencia fornecidas por uma equipa de especialistas. Existem 5 aptidoes :

a1-- habitacao

fornecer uma 'hard copy' do que foi digitalizado, assim como periférico de saída onde as respostas às perguntas podem ser dadas em mapas.

Os pontos digitalizados e os seus descritores são enviados do terminal gráfico para o computador central, o qual os armazena no disco em cláusulas unitárias.

O computador central, um PDP 11/23 da DEC endereça 128K bytes de memória central, e corre um interpretador Prolog sob o sistema de exploração RT11. Comunica com a unidade RX02 da DEC de 2 discos flexíveis de densidade dupla. O terminal é um VT100 da DEC. Está planeada a substituição a curto prazo da memória por 256 Kbytes, e dos discos flexíveis por um disco duro de tecnologia Winchester de 36,4 Mbytes. Este último permitirá a utilização do sistema de exploração UNIX e do interpretador Prolog correspondente.

ORGANIZACAO DO SISTEMA =====

O nosso sistema pericial compreende uma base de dados com factos e uma base de conhecimentos com regras de inferência, sendo capaz de responder a questões usando essas regras para raciocinar sobre os factos. Também explica tais raciocínios, com maior ou menor detalhe, de acordo com as necessidades do utilizador. Regras e factos podem ser actualizados pelo utilizador e novos itens serão automaticamente incorporados sem necessidade de reprogramação. As perguntas podem ser feitas quer em linguagem natural (Português) quer seguindo as opções de um menu. Podem dizer respeito ao domínio de conhecimento, incluindo a maneira como tal conhecimento está organizado, ou sobre as possibilidades de linguagem natural que o sistema possui, aliviando em consequência o utilizador da necessidade de consultar um manual.

Além disso a gramática compreende o tratamento das elipses, extraposição e a coordenação de frases e de relativas. Adicionalmente assegura grande número de testes quer sintácticos quer semânticos, que despoletam mensagens de erro apropriadas.

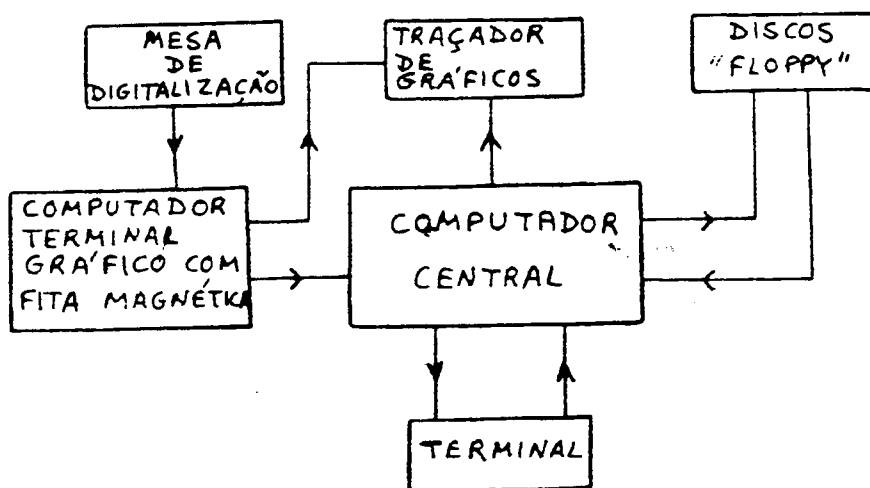
Ha' quatro programas separados : um em Basic para a digitalização através de uma leitora; outro em Pascal para receber os pontos digitalizados e armazená-los em disco; outro ainda em Prolog para actualização do conhecimento; e um que compreendendo três módulos básicos, todos em Prolog - a interface em lingua-

- a2-- agricultura nao intensiva
- a3-- agricultura intensiva
- a4-- industria
- a5-- infraestruturas de recreio

Finalmente uma resra sobre a sintese biofisica baseia-se nos valores e representatividade das aptidoes para extrair indicacoes de areas privilegiadas, de acordo com os objectivos de desenvolvimento.

O conhecimento dos especialistas e' tambem tornado explicito atraves de um grande numero de textos explicativos, um para cada valor possivel de cada descriptor.

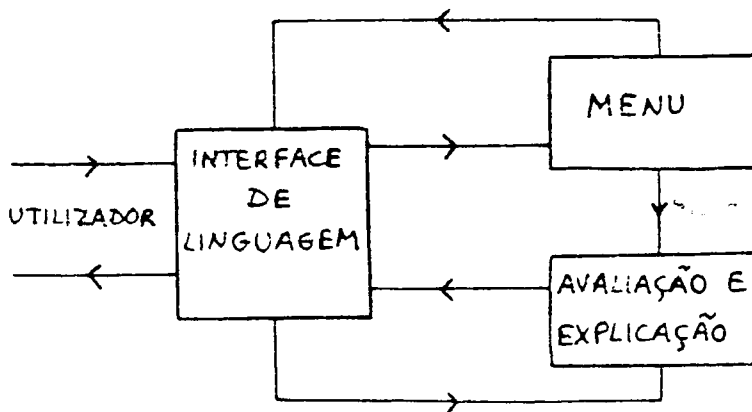
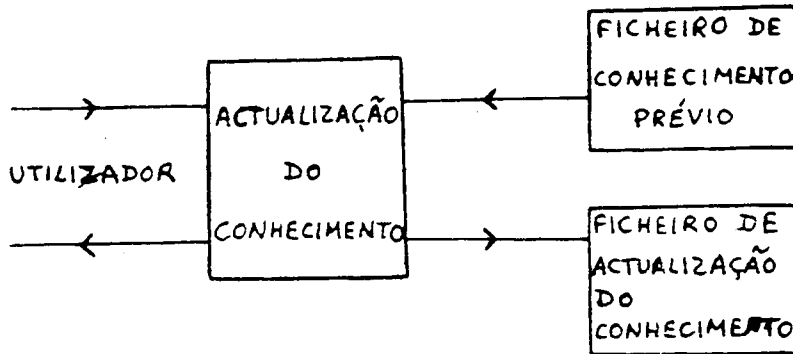
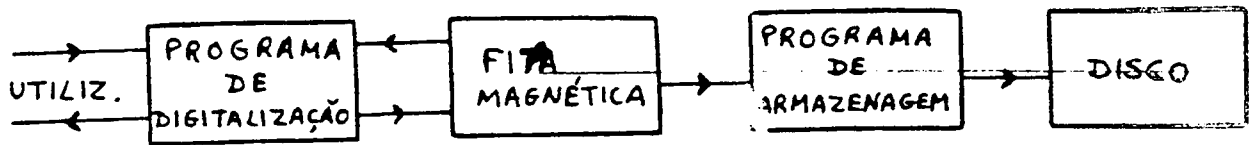
CONFIGURACAO DO SISTEMA
 =====



A digitalização e' feita 'off line' usando um digitalizador Tektronix 491 e o computador terminal grafico (com fita magnetica) Tektronix 4052. Para tal e' usada a linguagem Basic. O leitor interessado pode consultar [1] para detalhes. Basta dizer que o utilizador pode indicar que todos os pontos de uma area fechada tem o mesmo par valor/representatividade fornecendo apenas o seu contorno. O programa atribui esse mesmo par a todos os pontos da grelha dentro desse contorno. Qualquer conflito entre contornos e' figurado graficamente e uma decisao e' pedida ao utilizador.

O tracador de graficos, um Tektronix 4662, pode ser usado para monitorar

em natural, o menu e o modulo de avaliacao e explicacao- todos ligados como se mostra a seguir.



Organização dos Programas

Inicialmente o utilizador comunica com a interface em linguagem natural, seja para exprimir uma pergunta, um comando ou um facto, seja para dar a iniciativa ao programa enviando pelo teclado a palavra 'fala'. Neste caso o modulo menu é empregue para guiar o utilizador guiado através de um menu de questões possíveis ou ajudado na construção da sua pergunta. No entanto, a qualquer momento pode recuperar a iniciativa falando a palavra 'ouve' como resposta a qualquer pergunta do programa. Nesse caso a interface em linguagem natural é então de novo entrada. Em qualquer dos casos, depois de identificada a pergunta, o modulo de avaliacao e explicacao é accionado para produzir a

Durante a avaliação, faz-se o acesso à base de dados externa em disco e os pontos e seus descritores, e uma estrutura explicativa é gerada caso o utilizador a requeira.

Depois da resposta ser produzida no terminal e/ou desenhada no traçador gráfico, e uma explicação eventualmente pedida e recebida, o módulo de linguagem natural é reentrado ficando pronto para novas perguntas ou terminar a sessão.

O Programa de actualização do conhecimento lê a base de conhecimentos, interacciona com o utilizador para aceitar quaisquer actualizações, faz testes de forma e consistência e produz um ficheiro com informação actualizada. O utilizador poderá depois instruir o programa principal para os incorporar.

O meta-conhecimento é necessário para tal, permitindo ao programa responder a perguntas sobre si próprio, nomeadamente sobre as regras que existem sobre os componentes dessas regras, e sobre a organização e dependências dos vários conceitos do domínio de aplicação.

A competência linguística de ORBI é conseguida por meio da análise léxica e da análise sintática/semântica, que transformam uma frase em linguagem natural numa expressão lógica executável [3].

A análise léxica substitui cada palavra da frase de entrada pela correspondente categoria(s) léxica, dada por um dicionário. A análise sintática e semântica é realizada por meio de uma gramática programada contendo regras sensíveis ao contexto, e inclui testes de consistência sintática e semântica e respectivas mensagens de erro.

As regras lidam com as estruturas fundamentais do Português, nomeadamente:

- perguntas sim/não
- perguntas que/qual/quantos/onde/porque (pronomes interrogativos)
- comandos
- cláusulas afirmativas, negativas, relativas, preposicionais, coordenadas extrapostas e elididas
- qualificação, abreviação e complementação complexa de substantivos
- quantificação universal, existencial, cardinal, definida

- verbos e advérbios
- adjectivos
- terminologia metalinguística, permitindo interrogar o programa sobre a sua própria competência linguística.

Esta gramática é independente das aplicações, sendo transportável para outros domínios.

Ela é completada por um módulo dependente da aplicação contendo estruturas, vocabulário e controle semântico dependentes do domínio de ORBI.

Dedicamos muita energia à incorporação de estruturas elididas e extranxas na competência linguística de ORBI porque elas são essenciais a uma boa interacção natural. Uma interface de linguagem natural incapaz de perceber frases envolvendo estas estruturas não é verdadeiramente natural.

A descrição técnica linguística das estruturas de frases aceites e exemplos encontram-se em [3].

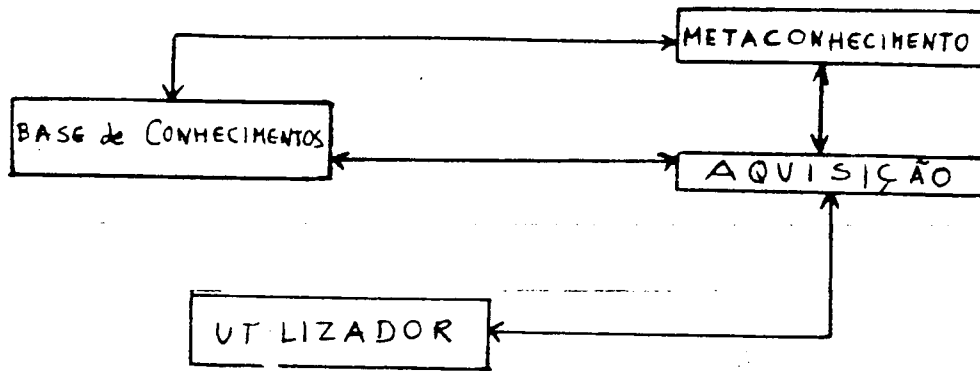
AQUISICAO DO CONHECIMENTO =====

A aplicação dos Sistemas Periciais a problemas dimensionados para o mundo real põe com toda a pertinência a necessidade de Bases de Conhecimentos dinâmicas.

É fundamental permitir ao utilizador não informático em interacção com o sistema a introdução de novo conhecimento de forma fácil, simples e segura, sem recorrer a qualquer reprogramação.

Novos conceitos e sobretudo novas regras podem vir a ser adicionadas harmonicamente ao conhecimento já incorporado, assim como a modificação de regras que, existindo anteriormente, se verificou serem erróneas ou incompletas.

ORBI utiliza para actualizar a sua Base de Conhecimentos um módulo individualizado do resto do sistema, o módulo de Aquisição, em íntima ligação com o módulo do Metaconhecimento.

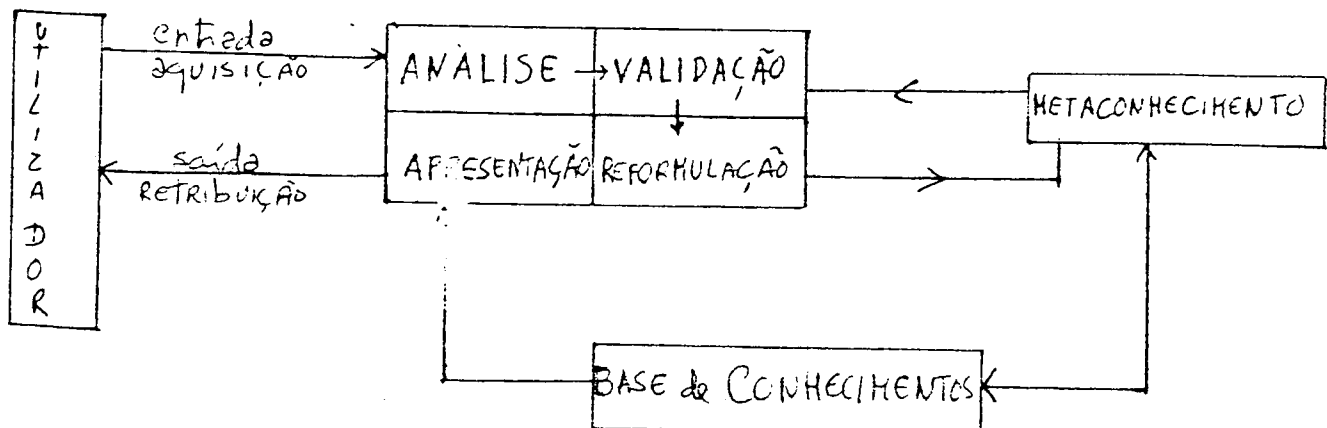


Um sistema de aquisição de conhecimentos pode caracterizar-se pela sua maior ou menor generalidade, pela sua maior ou menor 'especificidade', i.e. pela quantidade de diálogo necessário à aprendizagem, pela maior ou menor complexidade das estruturas de conhecimento passível de ser absorvido, pela sua capacidade de testar incoerências, etc.

Em GRBI o sistema de aquisição serve-se de uma rede semântica de conceitos estruturada a vários níveis que fornecem, entre outras indicações, todos os laços de dependência multidireccional de todos os conceitos e objectos inerentes ao domínio em causa.

Tal rede associativa, existente no módulo do Metaconhecimento, é ainda completada pelo modelo genérico sintático das regras onde está individualizado o raciocínio dedutivo dos especialistas.

É portanto este módulo que de certa forma valida a entrada. Qualquer adaptação a um novo domínio passaria necessariamente pela alteração dos valores nos nós desta rede associativa mantendo-se no entanto a sua estrutura geral.



AQUISIÇÃO E RETRIBUIÇÃO DE REGRAS

Se o utilizador pretender alterar uma regra já existente, esta é-lhe apresentada e, após a introdução da sua substituta, o sistema depois de reservar uma parte fixa do ecrã para fazer a apresentação do modelo genérico da regra em BNF (uma convenção para exprimir a sintaxe das regras) incorpora a que lhe é fornecida.

Durante este processo, vários testes, não só sintáticos como semânticos, são efectuados recorrendo ao metaconhecimento, sendo-se mensagens explicativas de rejeição caso se violem princípios de dependência conceptual ou então validando-se a regra introduzida.

Por exemplo, não será permitida a introdução de regras para factores que dependam de artidos ou cujos valores numéricos possíveis excedam os limites dos intervalos permitidos.

Uma diferença substancial entre ORBI e outros Sistemas Periciais que permitem a aquisição é a complexidade das suas regras de produção. Estas, na verdade estão longe da simplificação imposta às regras habituais de qualquer Sistema Pericial em que algumas premissas simples se ligam conjuntamente para a obtenção de uma conclusão.

ORBI tem a capacidade para admitir regras complexas resultantes da totalidade das condições necessárias e suficientes à obtenção das conclusões que podem, elas próprias, aparecer em alternativa.

Evitamos assim mais do que uma regra alternativa para o mesmo conceito, o que além de pouco natural poderá ser menos eficiente.

Suponhamos então que queremos dar entrada à seguinte regra para o cálculo da artido 1 'habitat concentrado', a qual depende selectivamente dos factores 4 (valores 4 ou 5), 6a (valor s ou 3), 6b (todos excepto 0, e, f), 6c (5), 6f (5), e no caso em que estas condições não se verificarem será a soma percentual ponderada dos factores 1 (peso 40), 2 (peso 25), 2a (peso 20), 2b (peso 5), 3 (peso 5) e 5 (peso 5).

O utilizador escreverá no terminal (segundo a BNF):

a1=0 se

f4=4 ou f4=5 ou f6a=s ou f6a=3 ou f6b\=0 e f6b\=e e f6b\=f

ou $f6c=5$ ou $f6f=5$

senao $a1=40*f1+25*f2+20*f2a+5*f2b+5*f3+5*f5$.

A regra e' lida como um termo e processada por um analizador que sucessivamente a decompoe atendendo a prioridade dos operadores pre-declarados.

Validadas as dependencias dos conceitos envolvidos a regra e' reformulada em termos de uma representacao interna ainda bastante legivel, (por utilizar operadores significativos do ponto de vista declarativo), mas ja' tambem procedimental. Note-se que a estrutura inicial e' mantida, o que e' fundamental para as respostas explicativas.

Caso o utilizador persunte pela regra, a sua representacao interna e' traduzida e retribuida exactamente na forma em que foi fornecida inicialmente.

Importante e' notar que toda a informacao procedimental adicional necessaria esta' implicita na regra, sendo por consequencia gerada automaticamente, poupando ao utilizador o fornecimento dessa informacao implicita.

Em conclusao, parece-nos possivel realocar o COREI quanto a aquisicao do conhecimento:

- a sua composicao modular,
- a interacao entre os modulos de aquisicao e metaconhecimento para efeitos de testes de validacao,
- o evitar de dialogos monos no processo de assimilacao de novo conhecimento,
- a retribuiçao das regras na forma exacta com que foram introduzidas,

e sobretudo

- a facilidade de absorver regras complexas mais proximas da formulacao do conhecimento dos especialistas no respectivo dominio sem grande tratamento previo ;
- a forma das regras e' muito geral, pois comporta condicionais (se, senao), conjuncoes (e), disjuncoes (ou), operadores relacionais e suas negacoes (=, \=, >=, =<, >, <)

- e expressões aritméticas,
- a representação interna das regras acompanha a estrutura original, de forma a possibilitar a produção de explicações de resultados que reflectem a estruturação do conhecimento fornecido pelo utilizador,
- o módulo de aquisição e apresentação de regras é pois portátil para outras aplicações.

Nos anexos abaixo discriminados apresentamos:

- 1- Exemplos de interrogação do sistema em português sobre uma região do caminho de Sesimbra
- 2- Exemplo de utilização do módulo de aquisição de novos conhecimentos e de interrogação do sistema sobre os seus conhecimentos e organização.

Listagem dos seguintes módulos:

- 3- Dicionário nuclear
- 4- Dicionário específico
- 5- Morfologia
- 6- Sintaxe nuclear
- 7- Semântica nuclear
- 8- Sintaxe e semântica específicas
- 9- Conhecimento metalinguístico
- 10- Avaliação, acesso à base de dados e produção de explicações e resultados
- 11- Metaconhecimento
- 12- Aquisição de novo conhecimento
- 13- Interrogação por menu
- 14- Base de conhecimentos
- 15- Mensagens do sistema
- 16- Digitalização
- 17- Controlador do traçador gráfico
- 18- Interfaces
- 19- Armazenagem dos dados digitalizados na base de dados
- 20- Base de dados com cerca de 600 pontos e seus descritores numa região

de Sesimbra.

AGRADECIMENTOS

=====

Agradecemos à Junta Nacional de Investisacao Cientifica pelo seu apoio financeiro adicional ao contracto com o Sevico de Estudos do Ambiente da Secretaria de Estado do Ambiente (contrato n. 313.81.52).

Os nossos agradecimentos vao tambem para A. Porto e M. Filsueiras pela ajuda na melhoria do sistema Prolos, a A.Porto pelo programa de acesso a base de dados, a V. Dias pelo programa de digitalizacao, e a tres alunos da licenciatura em engenharia informatica pela sua contribuicao para o controlador e tracador grafico.

REFERENCIAS

=====

[1] Dias,V. ; Pereira,L.M.

Digitalizacao de dados territoriais

Departamento de Informatica, 1982

[2] Oliveira,E.

A representacao do conhecimento num sistema pericial visto como uma base de dados logica

2 Congresso Portugues de Informatica, 1982

[3] Pereira,L.M. ; Oliveira,E. ; Sabatier,P.

ORBI-um sistema pericial para o ordenamento biofisico do territorio interogavel em Portugues

Dept. de Informatica U.N.L., Marco de 1982

(e tambem nas actas do 2. Congresso Portugues de Informatica)

[4] Pereira,L.M. ; Sabatier P. ; Oliveira E.

ORBI-an expert system for environmental resource evaluation through natural language

(versao em ingles de [3])

Dept. de Informatica U.N.L., Marco de 1982

(e tambem nas actas da First Logic Programming Conference, Marselha)

[5] Serviço de Estudos do Ambiente

Ordenamento biofísico do concelho de Sesimbra,
tratamento automático

Relatório preliminar, 1981

EXEMPLOS DE INTERROGAÇÃO DO SISTEMA EM PORTUGUÊS
SOBRE UMA REGIÃO DO CONCELHO DE SESIMBRA

Programa URBI : especialistas em ordenamento biofisico

Nucleo de Inteligencia Artificial
Departamento de Informatica
Universidade Nova de Lisboa

sempre que aparece o sinal barra (-) a iniciativa e' sua para escrever uma frase. Esta pode ser uma pergunta ou uma afirmacao.

Termine a frase com (.) ou (?) para uma resposta simples.

Termine com (+) para obter explicacoes nao detalhadas.

Termine com (++) para obter explicacoes detalhadas (incluindo descriçao).

Termine com (*) para obter o registo de sessao na impressora.

Pode combinar (*) com (+) ou (++).

Acabe sempre com (RETURN).

- quais são os descritores ?

descritores

- 1 equalidade dos microclimas
- 2 zonas climaticas homogeneas
- 3 permeabilidade do substracto
- 4 resistencia mecanica do substracto
- 5 estabilidade geodinamica do substracto
- 5a inclinacao das encostas
- 5b risco potencial de incendio corrido
- 5c recursos do sub-solo
- 6 sitios e elementos geologicos
- 6a arqueologia
- 7 escoamento pluvial superficial
- 8 potencial de aguas subterraneas
- 9 vulnerabilidade a poluicao dos mantos aquiferos
- 10 riscos de erosao
- 10a capacidade pedologica
- 11 equilibrio da vegetacao
- 12 resistencia vegetal
- 13 riquezas floristicas
- 14 potencialidade cinestetica e paisagistica
- 15a valores historicos construidos
- 15b potencial de recreio
- 16 areas degradadas
- 17 acessibilidade viaria

- e os factores ?

factor

- 1 conforto climático
- 2 capacidade dos terrenos 'a fundacao
- 2a potencial em aguas-abastecimento
- 2b resistencia 'a poluicao de mantos aquiferos
- 2c dinamica da vegetacao natural
- 3 resistencia dos solos 'a erosao
- 4 capacidade 'a producao agricola
- 5 resistencia ao incendio
- 6a riquezas geologicas
- 6b fitocenoses notaveis
- 6c riquezas culturais
- 6d paisagens desestabilizadas
- 6e riquezas do sub-solo
- 6f riquezas cinegeticas e paisagisticas
- 7 oferta do territorio para o recreio
- 8a acessibilidade do territorio

- e as atividades ?

atividade

- 1 habitacao
- 2 agricultura intensiva
- 3 agricultura nao intensiva
- 4 industria
- 5 infraestruturas de recreio

- quantos são os factores ?

16

- que pontos existem na regioao 91 , 93 ; 671 , 677 ?

ponto

91 671
91 673
91 675
91 677
93 671
93 673
93 675
93 677

- os pontos da resiao 93 , 95 ; 701 , 701 cujo descriptor 1 vale 2 ?

ponto

95 701

- Os pontos de resiao 91 , 95 ; 671 , 675 cujo f_2 e' maior que o f_1 ?

, ponto

95 673
93 675
93 673
93 671
91 675
91 675
91 671

- os pontos da rs 91 , 93 ; 673 , 675 com maximo d 2 ?

ponto

93 675

93 673

- existem pontos na rs 91 , 93 ; 671 , 675 ?

e' verdade

- que descriptors do P 91 , 671 tem o valor 2 ?

descriptor valor representatividade

1	2	5
4	2	5
9	2	5
10a	1	5
11	2	4
26	2	5

- quais os factores do # 91 , 671 com maior valor ?

factor valor representatividade

of 5 5

- e com menor valor ?

factor valor representatividade

ae	0	5
ed	0	5
ec	0	5

- os pontos da resisao 85 , 87 ; 701 , 707 com o mesmo d 4 ?

descriptor valor representatividade

4 0 3
ponto

85 701

85 703

4 2 4
ponto

85 705

87 701

87 705

4 2 5
ponto

85 707

87 707

4 2 3
ponto

87 703

- em cada ponto da resiso 95 , 87 ; 701 , 703 quais os descritores com ^{mesm} valores

ponto

95 701

valor representatividade

3 2
descriptor valor representatividade

1 3 2

1 4
descriptor valor representatividade

2 1 4

0 3
descriptor valor representatividade

3 0 3

4 0 3

5 0 3

2 2
descriptor valor representatividade

5a 2 2

5 5
descriptor valor representatividade

5b 5 5

8 5 5

0 4
descriptor valor representatividade

5c 0 4

2 3
descriptor valor representatividade

6 2 3

0 5
descriptor valor representatividade

6a 0 5

23a 0 5

26 0 5

1 5
descriptor valor representatividade

7 1 5

4 5
descriptor valor representatividade

9 4 5

2 1
descriptor valor representatividade
10 2 1

1 3
descriptor valor representatividade
10a 1 3

4 2
descriptor valor representatividade
11 4 2

5 2
descriptor valor representatividade
12 5 2

a 2
descriptor valor representatividade
13 a 2

5 3
descriptor valor representatividade
20 5 3

3 4
descriptor valor representatividade
23b 3 4

a 5
descriptor valor representatividade
24 a 5

85 703

valor representatividade

3 5
descriptor valor representatividade
1 3 5

1 5
descriptor valor representatividade
2 1 5
7 1 5

0 3
descriptor valor representatividade
3 0 3
4 0 3
5 0 3

5 4
descriptor valor representatividade
5a 5 4

1 1
descriptor valor representatividade
5b 1 1

0 5
descriptor valor representatividade
5c 0 5
6a 0 5
12 0 5
23a 0 5
26 0 5

2 3
descriptor valor representatividade
6 2 3

5 5
descriptor valor representatividade
8 5 5

4 5
descriptor valor representatividade
9 4 5

2 4
descriptor valor representatividade
10 2 4

1 3
descriptor valor representatividade
10a 1 3

4 2
descriptor valor representatividade
11 4 2

a 3
descriptor valor representatividade
13 a 3

4 3
descriptor valor representatividade
20 4 3

3 4
descriptor valor representatividade

23b 3 4

a 4
descriptor valor representatividade

24 a 4

87 701

valor representatividade

1 5
descriptor valor representatividade

1 1 5
7 1 5

1 3
descriptor valor representatividade

2 1 3

4 3
descriptor valor representatividade

3 4 3

2 4
descriptor valor representatividade

4 2 4

3 4
descriptor valor representatividade

5 3 4

5 5
descriptor valor representatividade

5a 5 5
5b 5 5
8 5 5

a 3
descriptor valor representatividade

5c a 3

2 1
descriptor valor representatividade

6 2 1

0 5
descriptor valor representatividade

6a 0 5
23a 0 5

4	5			
		descriptor	valor	representatividade
			9	4 5
5	3			
		descriptor	valor	representatividade
			10	5 3
			10a	5 3
			20	5 3
5	2			
		descriptor	valor	representatividade
			11	5 2
			12	5 2
			13	5 2
1	2			
		descriptor	valor	representatividade
			23b	1 2
			26	1 2
2	1			
		descriptor	valor	representatividade
			24	a 1

87 703

valor representatividade

4	4			
		descriptor	valor	representatividade
			1	4 4
			20	4 4
1	3			
		descriptor	valor	representatividade
			2	1 3
5	4			
		descriptor	valor	representatividade
			3	5 4
2	3			
		descriptor	valor	representatividade
			4	2 3
			10	2 3
			10a	2 3
3	4			

descriptor valor representatividade

5 3 4

5

5

descriptor valor representatividade

5a 5 5
8 5 5

1

2

descriptor valor representatividade

5b 1 2

a

3

descriptor valor representatividade

5c a 3

2

2

descriptor valor representatividade

6 2 2

0

5

descriptor valor representatividade

6a 0 5
23a 0 5

1

5

descriptor valor representatividade

7 1 5

4

5

descriptor valor representatividade

9 4 5

4

2

descriptor valor representatividade

11 4 2

5

2

descriptor valor representatividade

12 5 2

a

2

descriptor valor representatividade

13 a 2

3

2

descriptor valor representatividade

23b 3 2

a

1

descriptor valor representatividade

24 a 1

0 4

descriptor valor representatividade

26 0 4

- que pontos da rg 87 , 89 ; 671 , 675 tem o d 12 entre 2 e 4 ?

Ponto

87 673
87 675

- quantos pontos ha na regioao 85 , 89 ; 671 , 675 ?

9

— qual o f 4 de cada ponto da rs 85 , 89 ; 671 , 675 ?

ponto	factor	valor	representatividade
85 671	4	3	4
85 673	4	2	3
85 675	4	1	4
87 671	4	3	4
87 673	4	2	4
87 675	4	2	5
89 671	4	2	5
89 673	4	2	5
89 675	4	2	5

- e o r 5 ?

ponto factor valor representatividade

85	671	5	2	2
85	673	5	4	2
85	675	5	4	3
87	671	5	1	4
87	673	5	4	2
87	675	5	4	2
89	671	5	1	4
89	673	5	1	5
89	675	5	1	5

- os factores do p 39 , 673 ?

factor valor representatividade

1	1	5
2	3	5
2a	5	5
2b	4	4
2c	0	5
3	4	5
4	2	5
5	1	5
6a	0	5
6b	0	5
6c	0	5
6f	5	5
7	4	5
8a	2	4

- qual o valor da aptidão 3 do ponto 89 , 673 ++

ponto

89 673

a aptidão 3 - agricultura não intensiva tem o valor apto
(3 com representatividade 5) porque:
3 foi calculado por

$(25*(6-0)+10*4+5*4+60*2)/100$ por substituição em
 $(25*(6-f2c)+10*f3+5*f2b+60*f4)/100$

o factor 2b - resistência à poluição de mantos aquíferos tem o valor 4
(com representatividade 4) porque:

o descriptor 9 - vulnerabilidade à poluição dos mantos aquíferos tem o va
(com representatividade 4) porque:

Os aquíferos ocorrentes mais importantes apresentam uma protecção
ELEVADA

dado pertencerem ao manto Mio-Pliocénico cativo.

o factor 3 - resistência dos solos à erosão tem o valor 4
(com representatividade 5) porque:
4 foi calculado por

$(30*(6-1)+70*4)/100$ por substituição em
 $(30*(6-d7)+70*d10)/100$

o descriptor 7 - escoamento pluvial superficial tem o valor 1
(com representatividade 5) porque:

O escoamento pluvial superficial é
MUITO BAIXO

dado que as características da bacia hidrográfica onde se encontra incluído
e a pluviosidade média anual determinam um número de aspereza considerado
INFERIOR A 150

o descriptor 10 - riscos de erosão tem o valor 4
(com representatividade 5) porque:

Os solos são considerados como tendo riscos de erosão
BAIXO

dado apresentarem como limitações as associadas à sua reduzida espessura
efectiva.

o factor 4 - capacidade de produção agrícola tem o valor 2
(com representatividade 5) porque:

o descriptor 10a - capacidade pedológica tem o valor 2
(com representatividade 5) porque:

Os solos não são susceptíveis de utilização agrícola, salvo em casos
especiais, apresentando poucas ou moderadas limitações para pastagens,
exploração de matas e silvicultura. A aptidão pedológica à agricultura
pode ser considerada baixa.

o factor 2c - dinâmica da vegetação natural tem o valor 0
(com representatividade 5) porque:

o descriptor 11 - equilíbrio da vegetação tem o valor 0
(com representatividade 5) porque:

- e da aptidão 5 +

Ponto

89 673

a aptidão 3 - agricultura não intensiva tem o valor apto
(3 com representatividade 5) porque:
3 foi calculado por

$(25*(6-0)+10*4+5*4+60*2)/100$ por substituição em
 $(25*(6-f2c)+10*f3+5*f2b+60*f4)/100$

o factor 2b - resistência à poluição de mantos aquíferos tem o valor 4
(com representatividade 4), o factor 3 - resistência dos solos à erosão
(com representatividade 5) 4 foi calculado por *o valor*

$(30*(6-1)+70*4)/100$ por substituição em
 $(30*(6-d7)+70*d10)/100$

o factor 4 - capacidade de produção agrícola tem o valor 2
(com representatividade 5), o factor 2c - dinâmica da vegetação natural
(com representatividade 5) *o valor 0*

- quais os factores do ponto 85 , 707 que valem menos que 2 ?

nao ha

- quais os factores do ponto 85 , 707 que valem 5 ++

Ponto

85 707

o factor 2a - potencial em águas-abastecimento tem o valor 5
(com representatividade 5) porque:

o descriptor 8 - potencial de águas subterraneas tem o valor 5
(com representatividade 5) porque:

O potencial em águas subterraneas e'
MUITO ELEVADO
por se tratar de recursos do aquifero cativo do Mio-Pliocenico facilmente
exploraveis.

Ponto

85 707

o factor 2c - dinamica da vegetacao natural tem o valor 5
(com representatividade 3) porque:

o descriptor 11 - equilibrio da vegetacao tem o valor 5
(com representatividade 3) porque:

O equilibrio da vegetacao e' considerado
MUITO ELEVADO
devido ao facto das fitocenoses naturais se encontrarem num estado

SIMILAR
ao do climax potencial

Ponto

85 707

o factor 5 - resistencia ao incendio tem o valor 5
(com representatividade 2) porque:
5 foi calculado por

$(85 \times 5 + 15 \times 5) / 100$ por substituição em
 $(85 \times d12 + 15 \times d5b) / 100$

o descriptor 12 - piroresistencia vegetal tem o valor 5
(com representatividade 2) porque:

a Piro resistencia vegetal e'
MUITO ELEVADA
por se tratarem de formacoes onde :

- o grau de humidade do solo no periodo critico ;
- a biomassa combustivel ;
- a humidade relativa no interior da formacao ;
- e a combustibilidade directa das especies que a constituem

determinam um grau de combustibilidade da formacao MUITO BAIXO

o descriptor 5b - risco potencial de incendio corrido tem o valor 5
(com representatividade 5) porque:

O risco potencial de incendio corrido e'
BAIXO
devido a ocorrerem temperaturas maximas no estio (corridas com o gradiente
altimetrico e a exposicao das encostas)
INFERIOR a 27 graus C.
ou ocorrer um declive medio inferior a acentuado e/ou nao existir coberto
arboreo denso.

Ponto

85 707

o factor 6f - riquezas cineseticas e paisagisticas tem o valor 5
(com representatividade 4) porque:

o descriptor 20 - potencialidade cinesetica e paisagistica tem o valor 4
(com representatividade 4) porque:

A potencialidade cinesetica e paisagistica dependente
da sensibilidade visual da area e da probabilidade de ocorrencia de
especies animais com interesse cinesetico e'
ELEVADA

e que valem 4 ?

Ponto

85 707

o factor 2b - resistencia `a poluicao de mantos aquiferos tem o valor 4
(com representatividade 5) porque:

o descriptor 9 - vulnerabilidade `a poluicao dos mantos aquiferos tem o valor
(com representatividade 5) porque:

Os aquiferos ocorrentes mais importantes apresentam uma proteccao
ELEVADA
dado pertencerem ao manto Mio-Pliocenico cativo.

existe algum descriptor do ponto 91 , 671 cujo valor seja 0 ?

e' verdade

qual a representatividade do descriptor 2 do ponto 91 , 671 ?

representatividade

5

- e o valor ?

valor representatividade

3 5

- na resiao 93 , 97 ; 669 , 673 , quais os pontos que tem o mesmo d 2 ++

descriptor valor representatividade

2 3 5
ponto

93 669

o descriptor 2 - zonas climaticas homogeneas tem o valor 3
(com representatividade 5) porque:

O conjunto das caracteristicas termo pluviometricas indicam condicoes
MEDIANAMENTE
favoraveis 'a' instalacao humana definitiva

ponto

93 671

o descriptor 2 - zonas climaticas homogeneas tem o valor 3
(com representatividade 5) porque:

O conjunto das caracteristicas termo pluviometricas indicam condicoes
MEDIANAMENTE
favoraveis 'a' instalacao humana definitiva

descriptor valor representatividade

2 4 4
ponto

95 669

o descriptor 2 - zonas climaticas homogeneas tem o valor 4
(com representatividade 4) porque:

O conjunto das caracteristicas termo pluviometricas indicam condicoes
favoraveis 'a' instalacao humana definitiva

descriptor valor representatividade

2 4 5
ponto

97 669

o descriptor 2 - zonas climaticas homogeneas tem o valor 4

(com representatividade 5) Porque:

O conjunto das características termo pluviométricas indicam condições favoráveis à instalação humana definitiva

ponto

95 671

O descriptor 2 - zonas climáticas homogêneas tem o valor 4 (com representatividade 5) Porque:

O conjunto das características termo pluviométricas indicam condições favoráveis à instalação humana definitiva

ponto

97 671

O descriptor 2 - zonas climáticas homogêneas tem o valor 4 (com representatividade 5) Porque:

O conjunto das características termo pluviométricas indicam condições favoráveis à instalação humana definitiva

ponto

97 673

O descriptor 2 - zonas climáticas homogêneas tem o valor 4 (com representatividade 5) Porque:

O conjunto das características termo pluviométricas indicam condições favoráveis à instalação humana definitiva

descriptor valor representatividade

2 4 2

ponto

93 673

O descriptor 2 - zonas climáticas homogêneas tem o valor 4 (com representatividade 2) Porque:

O conjunto das características termo pluviométricas indicam condições

favoraveis `a instalacao humana definitiva

descriptor valor representatividade

2 4 3
ponto

95 673

o descriptor 2 - zonas climaticas homogeneas tem o valor 4
(com representatividade 3) porque:

O conjunto das caracteristicas termo pluviometricas indicam condicoes
favoraveis `a instalacao humana definitiva

- as aptidoes do ponto 91 , 671 +

ponto

91 671

a aptidao 1 - habitacao tem o valor nao apto
(0 com representatividade 5) porque:
o factor af - riquezas cinesticas e paisagisticas tem o valor 5
(com representatividade 5)

ponto

91 671

a aptidao 2 - agricultura intensiva tem o valor apto
(3 com representatividade 5) porque:
3 foi calculado por

$(50*2+25*4+10*4+10*(6-2)+5*2)/100$ por substituiçao em
 $(50*f4+25*f2a+10*f3+10*(6-f2c)+5*f2b)/100$

o factor 4 - capacidade 'a producao agricola tem o valor 2 é 4
(com representatividade 5).o factor 2a - potencial em aguas-abastecimento
(com representatividade 5).o factor 3 - resistencia dos solos 'a erosao
(com representatividade 5) 4 foi calculado por é 4

$(30*(6-3)+70*4)/100$ por substituiçao em
 $(30*(6-d7)+70*d10)/100$

o factor 2c - dinamica da vegetacao natural tem o valor 2 aqui
(com representatividade 4).o factor 2b - resistencia 'a poluiçao de mantos é 2
(com representatividade 5)

ponto

91 671

a aptidao 3 - agricultura nao intensiva tem o valor apto
(3 com representatividade 5) porque:
3 foi calculado por

$(25*(6-2)+10*4+5*2+60*2)/100$ por substituiçao em
 $(25*(6-f2c)+10*f3+5*f2b+60*f4)/100$

o factor 2b - resistencia 'a poluiçao de mantos aquiferos tem o valor 2
(com representatividade 5).o factor 3 - resistencia dos solos 'a erosao é 4
(com representatividade 5) 4 foi calculado por

$(30*(6-3)+70*4)/100$ por substituiçao em
 $(30*(6-d7)+70*d10)/100$

o factor 4 - capacidade 'a producao agricola tem o valor 2
(com representatividade 5).o factor 2c - dinamica da vegetacao natural é 2
(com representatividade 4)

ponto

91 671

- o d 4 nao tem o valor 2 no p 91 , 671 .

nao e' verdade

- quantos são os descritores do ponto 91 , 671 cujo valor é 0 ?

descritores numero

5

- e cujo valor e' 1 ?

descriptores numero

0

- e cujo valor e' 2 ?

descriptores numero

6

Qual é a média dos descritores do ponto 93 , 701 ?

media

2

- qual o maior factor do ponto 93 , 701 ?

factor Valor representatividade

4	5	1
2a	5	5

— que factor do ponto 93 , 701 e' o maior ?

factor valor representatividade

4	5	1
2a	5	5

- no ponto 93 , 671 qual o factor 5 ++

Ponto

93 671

o factor 5 - resistencia ao incendio tem o valor 4
(com representatividade 4) porque:
4 foi calculado por

$(85*4+15*5)/100$ por substituição em
 $(85*d12+15*d5b)/100$

o descriptor 12 - Piroresistencia vegetal tem o valor 4
(com representatividade 4) porque:

a Piro resistencia vegetal e'
ELEVADA
por se tratarem de formacoes onde :

- o grau de humidade do solo no periodo critico ;
- a biomassa combustivel ;
- a humidade relativa no interior da formacao ;
- e a combustibilidade directa das especies que a constituem

determinam um grau de combustibilidade da formacao BAIXO.

o descriptor 5b - risco potencial de incendio corrigido tem o valor 5
(com representatividade 5) porque:

O risco potencial de incendio corrigido e'
BAIXO

devido a ocorrerem temperaturas maximas no estio (corrigidas com o gradiente
altimetrico e a exposicao das encostas)
INFERIOR a 27 graus C.
ou ocorrer um declive medio inferior a acentuado e/ou nao existir coberto
arboreo denso.

- e no ponto 91 , 671 ++

ponto

91 671

O factor 5 - resistencia ao incendio tem o valor 4
(com representatividade 4) porque:
4 foi calculado por

$(85*4+15*5)/100$ por substituicao em
 $(85*d12+15*d5b)/100$

O descriptor 12 - piroresistencia vesetal tem o valor 4
(com representatividade 4) porque:

O piro resistencia vesetal e'
ELEVADA

por se tratarem de formacoes onde :

- o grau de humidade do solo no periodo critico ;
- a biomassa combustivel ;
- a humidade relativa no interior da formacao ;
- e a combustibilidade directa das especies que a constituem

determinam um grau de combustibilidade da formacao BAIXO.

O descriptor 5b - risco potencial de incendio corrido tem o valor 5
(com representatividade 5) porque:

O risco potencial de incendio corrido e'
BAIXO

devido a ocorrerem temperaturas maximas no estio (corridas com o gradiente
altimetrico e a exposicao das encostas)
INFERIOR a 27 graus C,
ou ocorrer um declive medio inferior a acentuado e/ou nao existir coberto
arboreo denso.

- quais os factores da aptidão 1?

factor

- 1 conforto climatico
- 2 capacidade dos terrenos 'a fundacao
- 2a potencial em aguas-abastecimento
- 2b resistencia 'a poluicao de mantos aquiferos
- 3 resistencia dos solos 'a erosao
- 5 resistencia ao incendio

- e os descriptors ?

descriptor

- 1 qualidade dos microclimas
- 2 zonas climaticas homogeneas
- 4 resistencia mecanica do substrato
- 5a inclinacao das encostas
- 3 permeabilidade do substrato
- 5 estabilidade geodinamica do substrato
- 8 potencial de 'guas subterraneas
- 9 vulnerabilidade 'a poluicao dos mantos aquiferos
- 7 escoamento pluvial superficial
- 10 riscos de erosao
- 12 piroresistencia vegetal
- 5b risco potencial de incendio corrigido

- e do factor 5 ?

descriptor

12 Piroresistencia vegetal
5b risco potencial de incendio corrido

- de-me a resra da aptidao 5 .

resra

a5=0 se

f4=4

ou

f4=5

ou

f6a=s

ou

f6a=3

ou

f6c=5

senao

v.int=70*f7+20*f1+10*f2

a5=v.int*9/10/100

se

f6f=5

ou

f6b\=0 e f6b\=e e f6b\=f

senao

a5=v.int/100

- qual a resra de cada factor da aptidao 3 ?

factor resra

 2b resistencia `a poluicao de mantos aquiferos
f2b=d9

 3 resistencia dos solos `a erosao
f3=(30*(6-d7)+70*d10)/100

 4 capacidade `a producao agricola
f4=d10a

 2c dinamica da vesetacao natural
f2c=d11

- e da aptidão 4 ?

resra

a4=0 se

f4=4

ou

f4=5

ou

f6a=s

ou

f6a=3

ou

f6b\=0 e f6b\=e e f6b\=f

ou

f6c=5

ou

f6f=5

seriao

v.int=30*f2+30*f2b+15*f2a+15*f3+10*f5

f6e=0 e a4=v.int/100

ou

f6e=a e a4=v.int*3/4+50/100

- e de cada factor da aptidao 4 ?

factor resra

2 capacidade dos terrenos 'a fundacao
 $f2=(65*d4+15*d5a+10*d3+10*d5)/100$

2b resistencia 'a poluicao de mantos aquiferos
 $f2b=d9$

2a potencial em aguas-abastecimento
 $f2a=d8$

3 .resistencia dos solos 'a erosao
 $f3=(30*(6-d7)+70*d10)/100$

5 resistencia ao incendio
 $f5=(85*d12+15*d5b)/100$

oe riquezas do sub-solo

$f6e=a$ se

$d11=3$ e $d5c=a$ e
senao

$f6e=0$

- que regras das aptidoes existem ?

aptidao regra

1 habitacao

a1=0 se
 f4=4
 ou
 f4=5
 ou
 f6a=s
 ou
 f6a=3
 ou
 f6b\=0 e f6b\=e e f6b\=f
 ou
 f6c=5
 ou
 f6f=5

senao

$$a1=(40*f1+25*f2+20*f2a+5*f2b+5*f3+5*f5)/100$$

2 agricultura intensiva

a2=0 se
 f6a=s
 ou
 f6b\=0 e f6b\=f
 ou
 f6c=5

senao

$$a2=(50*f4+25*f2a+10*f3+10*(6-f2c)+5*f2b)/100$$

3 agricultura nao intensiva

a3=0 se
 f6a=s
 ou
 f6b=s
 ou
 f6c=s

senao

$$a3=(25*(6-f2c)+10*f3+5*f2b+60*f4)/100$$

4 industria

a4=0 se
 f4=4
 ou
 f4=5
 ou
 f6a=s
 ou
 f6a=3

```

    ou
    f6b\=0 e f6b\=e e f6b\=f
    ou
    f6c=5
    ou
    f6f=5
senao
v.int=30*f2+30*f2b+15*f2a+15*f3+10*f5
    f6e=0 e a4=v.int/100

ou
    f6e=a e a4=v.int*3/4+50/100

```

5 infraestruturas de recreio

```

a5=0 se
    f4=4
    ou
    f4=5
    ou
    f6a=s
    ou
    f6a=3
    ou
    f6c=5
senao
v.int=70*f7+20*f1+10*f2

a5=v.int*9/10/100

se
    f6f=5
    ou
    f6b\=0 e f6b\=e e f6b\=f
senao
a5=v.int/100

```


- quais as regras das aptidoes ?

aptidao regra

1 habitacao

a1=0 se
 f4=4
 ou
 f4=5
 ou
 f6a=s
 ou
 f6a=3
 ou
 f6b\=0 e f6b\=e e f6b\=f
 ou
 f6c=5
 ou
 f6f=5
senao

$$a1=(40*f1+25*f2+20*f2a+5*f2b+5*f3+5*f5)/100$$

2 agricultura intensiva

a2=0 se
 f6a=s
 ou
 f6b\=0 e f6b\=f
 ou
 f6c=5
senao

$$a2=(50*f4+25*f2a+10*f3+10*(6-f2c)+5*f2b)/100$$

3 agricultura nao intensiva

a3=0 se
 f6a=s
 ou
 f6b=s
 ou
 f6c=s
senao

$$a3=(25*(6-f2c)+10*f3+5*f2b+60*f4)/100$$

4 industria

a4=0 se
 f4=4
 ou
 f4=5
 ou
 f6a=s
 ou
 f6a=3

```

    ou
    f6b\=0 e f6b\=e e f6b\=f
    ou
    f6c=5
    ou
    f6f=5
senao
v.int=30*f2+30*f2b+15*f2a+15*f3+10*f5
    f6e=0 e a4=v.int/100

ou
    f6e=a e a4=v.int*3/4+50/100

```

5 infraestruturas de recreio

```

a5=0 se
    f4=4
    ou
    f4=5
    ou
    f6a=s
    ou
    f6a=3
    ou
    f6c=5
senao
v.int=70*f7+20*f1+10*f2

a5=v.int*9/10/100

se
    f6f=5
    ou
    f6b\=0 e f6b\=e e f6b\=f
senao
a5=v.int/100

```

o valor da aptidão 3 do ponto 91,707?
o valor da aptidão 3 dos pontos?
o valor das aptidões do ponto 91,707?
o valor das aptidões dos pontos?

5.

os valores das aptidões dos pontos?
o valor da aptidão 3 de cada ponto?
o valor de cada aptidão dos pontos?
o valor das aptidões de cada ponto?
o valor de cada aptidão de cada ponto?

10.

os valores de cada aptidão de cada ponto?
cada valor de cada aptidão do ponto 91,707?
o valor do D10 do f3 da A5 do P 91,707 da rs 95,705;90,710?
o valor de cada descriptor de cada factor de cada aptidão de cada ponto ?
o valor dos descriptors de cada factor da aptidão a3 do ponto 91,707?

15.

o valor dos descriptors de cada factor da aptidão 3 para o ponto 91,707?
o valor do d20 do P 91,707?
o valor do d 20 da a3 do P 91,707 da rs 95,707;90,710?
o valor do F3 do P 91,707?
o valor do f3 da A3 do P 91,707 da rs 95,707;90,710?

20.

o valor do D26 do F8 do P 91,707?

os pontos com valor 5 para a aptidão 3?
o valor das aptidões dos pontos com valor 5 para cada descriptor do factor?
o valor de cada descriptor dos pontos com valor 5 para cada aptidão?
cada ponto com valor 5 para cada descriptor de cada aptidão com valor 4
para cada factor?

5.

os pontos que tem a aptidão 3 igual a 3?
os pontos com a aptidão 3 igual a 5?
os pontos cuja a 3 e' igual a 5?
os pontos cujo descriptor 3 e' igual a 5?
o valor das aptidões de cada ponto cujo descriptor 8 de cada factor
e' igual a 5?

10.

o valor dos factores de cada aptidão dos pontos cujos factores são iguais a
os pontos que tem cada aptidão com valor 5?
os pontos cujas aptidões tem o valor 5?
os valores dos factores dos pontos com valor 5 para cada descriptor?
os valores dos factores dos pontos que tem cada descriptor com valor 5?

15.

quanto vale cada descriptor dos factores de cada aptidão dos pontos?
quais são os factores?
quais são os factores cujo valor de cada descriptor e' igual a 5?
quais os valores de cada aptidão que cada ponto tem?
quais são os pontos cuja media dos valores das aptidões e' igual a 2?

20.

que pontos tem a a3 com valor muito apto?
que valor a aptidão A3 de cada ponto cujo f2 e' 5 tem?
quantos pontos tem a a3 com o valor muito apto?
quantos factores de cada aptidão existem?
quantos pontos que tem 2 aptidões com valor não apto existem?

25.

cada aptidão dos pontos com valor 5 em cada factor e' entre 1 e 3?
será que alguns pontos cuja aptidão A3 e igual a 5 tem o f3 com o valor 4?

Quais são os factores da aptidão A3 no P 91,707?
Quais são os valores dos factores da aptidão A3 no P 91,707?
Quais são os pontos que têm o mesmo valor para a aptidão a3?
Da-me os pontos que têm o mesmo valor para cada aptidão!

5.

Quantos são os pontos que têm o mesmo valor da aptidão 3?
Quantos pontos que têm o mesmo valor para cada aptidão existem?
Será que alguns pontos que têm o mesmo valor para a aptidão a3 têm o factor
f3 entre 2 e 5?

Da-me os pontos cujo factor 3 é entre 2 e 5 e que têm o mesmo valor para
a a3!

A média dos valores da aptidão 3 dos pontos cujo factor 5 é superior a 3!

10.

Quais os pontos que têm o f3 superior a 3 e cuja média das aptidões é
entre 2 e 5?

Os pontos que têm cada descriptor superior a 2?

Qual é a média dos valores da aptidão 3 dos pontos que têm o descriptor 12
entre 2 e 5, que têm o factor 6 superior a 3 e que têm o f10 igual a 1?

A média dos valores da a3 dos pontos que têm o d12 entre 2 e 5, com o f6
superior a 3 e o f6 igual a 1!

A media dos valores da a_3 dos pontos que tem o d_{12} entre 2 e 5, com o f_6 superior a 3 e com o f_6 igual a 1.

Quais são as aptidoes que existem ?

Que aptidoes tem os pontos ?

Quais os pontos cuja a_3 não é 4 ?

Que aptidoes existem ?

5.

Existem pontos ?

Quantos pontos tem o valor 3 na aptidao 3 ?

Quais são as aptidoes no ponto 91,707 que têm o valor 4 ?

Que pontos tem a_3 igual a 3 ?

Quantos pontos cuja a_3 é 4 existem.

10.

Que pontos existem com a_3 igual a 4 ?

Quais são as aptidoes que no ponto 91,707 que o valor 3 ?

No ponto 91,707 quais são as aptidoes ?

Os pontos que não tem a a3 com valor muito apto !

Os pontos cuja a3 não é igual a 3?

O valor das aptidoes de cada ponto que não tem nenhum descriptor maior que

Os pontos que não tem nenhuma aptidão com valor muito apto ?

5.

Será que cada ponto cuja aptidão 3 não é igual a 2 não tem nenhum factor de valor 4?

Será que os pontos que não tem nenhuma aptidão inferior a 2 não tem nenhum descriptor superior a 2 ?

Quais são os factores dos pontos iguais a 5 ?
E as aptidoes ?
E as aptidoes do ponto cujo D8 é 5 ?

Quanto vale o F3 do ponto cujo D8 é igual a 5 ?
E a A3 ?
E a A3 de cada ponto cujo D12 é entre 2 e 4 ?

Que aptidão dos pontos cujo D8 é 3 vale 5 ?
E que factor ?
E que descriptors do ponto cujo F3 é 4 ?

Quantas aptidoes dos pontos cujo D8 é 3 valem 5 ?
E quantos factores ?
E quantos descriptors do ponto cujo F3 é 4 ?

Quantas são as aptidoes dos pontos cujo D8 é 3 ?
E os descriptors ?
E os factores do ponto cujo F3 é 4 ?

De-me as aptidoes dos pontos cujo D8 é 3 ?
E os factores ?
E os descriptors do ponto cujo F3 é 4 ?

Será que as aptidoes dos pontos cujo D8 é 3 valem 5 ?
E os factores ?
E os factores de cada ponto ?

Será que as aptidoes dos pontos cujo D8 é 3 são entre 1 e 4 ?
E valem 3 ?

Quais são os descriptors do factor 3 no ponto 91,707 ?
E no ponto 101,105 ?
E no ponto cuja aptidão 3 vale 5 ?

Quanto vale cada descriptor da aptidão 3 no ponto 91,707 ?
E no ponto 101,105 ?
E no ponto cujo factor 6 é entre 3 e 5 ?

Que aptidão do ponto 91,707 é superior a 3 ?
E do ponto 101,707 ?
E do ponto cujo factor 6 tem o valor 2 ?

Quantos factores da aptidão 5 no ponto 91,707 são superiores a 2 ?
E no ponto 101,105 ?

Quantas são as aptidoes do ponto 91,707 superiores a 2 ?
E do ponto 101,105 ?

De-me as aptidoes do ponto 91,707 ?
E do ponto 101,105 ?

Será que a aptidão 2 no ponto 91,707 tem o valor muito apto ?
E no ponto 101,105 ?
E nos pontos cuja aptidão 3 tem o valor não apto ?

Em cada ponto, quais são os factores de cada entidade superiores a 2?
Quais são, em cada ponto, os factores de cada entidade superiores a 2?
Quais são os factores de cada entidade em cada ponto superiores a 2?

Em cada ponto, quanto vale o factor de cada entidade?
Quanto vale, em cada ponto, o factor de cada entidade?
Quanto vale o factor de cada entidade em cada ponto?

Em cada ponto, que factores de cada entidade são superiores a 2?
Que factores de cada entidade em cada ponto são superiores a 2?

Em cada ponto, quantos factores de cada entidade são superiores a 2?
Quantos factores de cada entidade em cada ponto são superiores a 2?

Em cada ponto, quantos são os factores de cada entidade superiores a 2?
Quantos são, em cada ponto, os factores de cada entidade superiores a 2?
Quantos são os factores de cada entidade em cada ponto superiores a 2?

Para cada ponto, dá-me os factores de cada entidade superiores a 2?
Dá-me, para cada ponto, os factores de cada entidade superiores a 2?
Dá-me os factores de cada entidade para cada ponto superiores a 2?

Para cada ponto, os factores de cada entidade superiores a 2?
Os factores de cada entidade para cada ponto superiores a 2?

Para cada ponto, será que os factores de cada entidade são superiores a 2?
Será que, para cada ponto, os factores de cada entidade são superiores a 2?
Será que os factores de cada entidade para cada ponto são superiores a 2?

Para cada ponto, os factores de cada entidade são superiores a 2?
Os factores de cada entidade para cada ponto são superiores a 2?

EXEMPLO DE UTILIZAÇÃO DO MÓDULO
DE AQUISIÇÃO

EXEMPLOS DE AQUISIÇÃO E CONSULTA DO CONHECIMENTO

qual e' a palavra chave?eco

AQUISICAO DE REGRAS

A regra a modificar e' Ja' existente ou e' n)ova?

n

A sua regra permitira calcular conceitos de que tipo?!

a)aptidao; f)actor

a

Qual o numero de tal aptidao
se tiver duvidas chame meta)conhecimento.

meta

Os conceitos fundamentais sao:

s-sintese

a-aptidao

r-rede

f-factor

d-descriptor

p-ponto

Escreva as letras que lhe interessam separadas por espaco

a

Sobre cada um deles conheco:

1-a quantidade.

2-nomes das suas instanciaco es particulares.

3-de que dependem.

4-Para o que contribuem.

Escreva os numeros correspondentes ao que lhe interessa conhecer,
separados por espaco.

1 2 3 4

***aptidao

*5

*1-HABITACAO.

2-AGRICULTURA INTENSIVA.

3-AGRICULTURA NAO INTENSIVA.

4-INDUSTRIA.

5-INFRAESTRUTURAS DE RECREIO.

*factor

.xsintese

A regra a modificar e' Ja' existente ou e' n)ova?

n

A sua regra permitira calcular conceitos de que tipo?!

a)aptidao; f)actor

a

Qual o numero de tal aptidao
se tiver duvidas chame meta)conhecimento.

9

Nome da regra?

novaptidao

Use a seguinte sintaxe:

```
<REG> ::= <ALT> senao <COR> | <ALT> | <COR>
<ALT> ::= <IGU> se <CND> | <IGU>
<CND> ::= <DIJ> | <CJC> | <ALT>
<IGU> ::= <CON> = <VAL> | <CON> = <CON>
<COR> ::= <EXP> | <ALT> | <REGRA>
<CON> ::= <letra minuscula> <ALF>
<DIJ> ::= <CJC> ou <CJC> | <CJC>
<ALF> ::= <numero> | <numero> <letra minuscula>
<CJC> ::= <IGU> e <IGU> | <IGU>
<VAL> ::= <numero> | <numero> <letra minuscula>
<EXP> ::= <CON> = <PAR> + <PAR> | <CON> = <PAR> | <REGRA>
<PAR> ::= <numero> * <CON> | <numero> | <CON>
```

Escreva a regra com letras minusculas. Nomeie o valor desta regra por a9
Termine por ponto (.)

a9=0 se f2=1 ou f3\=0 e f3\=3 senao a9=30*f5+70*f2.

Posso memorizar a nova regra? sim ou nao?

sim

Mais alteracoes s/n?

s

A regra a modificar e' ja' existente ou e' nova?

e

A sua regra permitira calcular conceitos de que tipo?:

a) Aptidao; f)actor

a

Qual o numero de tal aptidao

se tiver duvidas chame meta)conhecimento.

1

aptidao1-HABITACAO.

a1=0 se

f4=4

ou

f4=5

ou

f6a=s

ou

f6a=3

ou

f6b\=0 e f6b\=e e f6b\=f

ou

f6c=5

ou

f6f=5

senao

a1=(40*f1+25*f2+20*f2a+5*f2b+5*f3+5*f5)/100

Use a seguinte sintaxe:

```
<REG> ::= <ALT> senao <COR> | <ALT> | <COR>
```

```
<ALT> ::= <IGU> se <CND> | <IGU>
```

```
<CND> ::= <DIJ> | <CJC> | <ALT>
```

```
<IGU> ::= <CON> = <VAL> | <CON> = <CON>
```

<COR> ::= <EXP> | <ALT> | <REGRA>
 <CON> ::= <letra minuscula> <ALF>
 <DIJ> ::= <CJC> ou <CJC> | <CJC>
 <ALF> ::= <numero> | <numero> <letra minuscula>
 <CJC> ::= <IGU> e <IGU> | <IGU>
 <VAL> ::= <numero> | <numero> <letra minuscula>
 <EXP> ::= <CON> = <PAR> + <PAR> | <CON> = <PAR> | <REGRA>
 <PAR> ::= <numero> * <CON> | <numero> | <CON>

Escreva a regra com letras minusculas. Nomeie o valor desta regra por a1
Termine por ponto (.)

a0=0 se f2=3 ou f6a=0.

errou ao nomear o valor da sua regra por a0

Escreva a regra com letras minusculas. Nomeie o valor desta regra por a1
Termine por ponto (.)

a1=5 se f6a=5 ou f4=3 senao a1=20*f1+60*f5+1.

posso memorizar a nova regra? sim ou nao?

nao

Nova tentativa, s/n?

s

A regra a modificar e' ja' existente ou e' nova?

n

A sua regra permitira calcular conceitos de que tipo?:

a) aptidao; f) factor

f

Qual o numero de tal factor

se tiver duvidas chame meta) conhecimento.

meta

Os conceitos fundamentais sao:

s-sintese

a-aptidao

r-rede

f-factor

d-descriptor

p-ponto

Escreva as letras que lhe interessam separadas por espaco

f

Sobre cada um deles conheco:

1-a quantidade.

2-nomes das suas instanciaco es particulares.

3-do que dependem.

4-para o que contribuem.

Escreva os numeros correspondentes ao que lhe interessa conhecer,
separados por espaco.

1 2 3 4

***factor

*16

*1-conforto climatico.

2-aptidao dos terrenos 'a fundacao.

2a-potencial em aguas-abastecimento.

2b-resistencia 'a poluicao de mantos aquiferos.

2c-dinamica da vegetacao natural.

3-resistencia dos solos 'a erosao.

4-aptidao 'a producao agricola.

5-resistencia ao incendio.
6a-riquezas geologicas.
6b-fitocenoses notaveis.
6c-riquezas culturais.
6d-paisagens desestabilizadas.
6e-riquezas do sub-solo.
6f-riquezas cineceticas e paisagisticas.
7-oferta do territorio para o recreio.
8a-acessibilidade do territorio.
*descriptor
*aptidao

A regra a modificar e' ja' existente ou e' nova?
n

A sua regra permitira calcular conceitos de que tipo?:
a)aptidao; f)factor
f

Qual o numero de tal factor
se tiver duvidas chame metaconhecimento.
8h
Nome da regra?
novofactor

Use a seguinte sintaxe:

<REG> ::= <ALT> senao <COR> | <ALT> | <COR>
<ALT> ::= <IGU> se <CND> | <IGU>
<CND> ::= <DIJ> | <CJC> | <ALT>
<IGU> ::= <CON> = <VAL> | <CON> = <CON>
<COR> ::= <EXP> | <ALT> | <REGRA>
<CON> ::= <letra minuscula> <ALF>
<DIJ> ::= <CJC> ou <CJC> | <CJC>
<ALF> ::= <numero> | <numero> <letra minuscula>
<CJC> ::= <IGU> e <IGU> | <IGU>
<VAL> ::= <numero> | <numero> <letra minuscula>
<EXP> ::= <CON> = <PAR> + <PAR> | <CON> = <PAR> | <REGRA>
<PAR> ::= <numero> * <CON> | <numero> | <CON>

Escreva a regra com letras minusculas. Nomeie o valor desta regra por f8h
Termine por ponto (.)
f8h=s se a1=0 ou d3=2.

A sua regra refere-se a factor. Logo nao depende de aptidao
Escreva a regra com letras minusculas. Nomeie o valor desta regra por f8h
Termine por ponto (.)
f8h=s se d6=0 ou d4=1 senao f8h=0 se d1\=0 senao f8h=20*d2+60*d3+2.
Posso memorizar a nova regra? sim ou nao?

sim
Mais alteracoes s/n?

n
Quer ver a regra? s/n

n
Com vista na actualizacao do metaconhecimento diga se as modificacoes
sao definitivas. Escreva sim ou nao.

Quer consultar o metaconhecimento? s/n

METACONHECIMENTO

O conhecimento sobre o dominio existe em REGRAS que trabalham sobre valores ligados a varios CONCEITOS. Para cada conceito conheço o numero e nome das suas instancias e as ligacoes a outros conceitos. Conforme o que desejar saber pode escrever uma das palavras sublinhadas. Nao lhe interessando o metaconhecimento escreva NAO.

conceitos

Os conceitos fundamentais sao:

s-sintese

a-aptidao

r-rede

f-factor

d-descriptor

p-ponto

Escreva as letras que lhe interessam separadas por espaco

a f

Sobre cada um deles conheço:

1-a quantidade.

2-nomes das suas instancias particulares.

3-do que dependem.

4-para o que contribuem.

Escreva os numeros correspondentes ao que lhe interessa conhecer, separados por espaco.

1 2

***aptidao

*6

*9-novaptidao

1-HABITACAO.

2-AGRICULTURA INTENSIVA.

3-AGRICULTURA NAO INTENSIVA.

4-INDUSTRIA.

5-INFRAESTRUTURAS DE RECREIO.

***factor

*17

*8h-novofactor

1-conforto climatico.

2-aptidao dos terrenos 'a fundacao.

2a-potencial em aguas-abastecimento.

2b-resistencia 'a poluicao de mantos aquiferos.

2c-dinamica da vegetacao natural.

3-resistencia dos solos 'a erosao.

4-aptidao 'a producao agricola.

5-resistencia ao incendio.

6a-riquezas geologicas.

6b-fitocenoses notaveis.

6c-riquezas culturais.

6d-paisagens desestabilizadas.

6e-riquezas do sub-solo.

6f-riquezas cineceticas e paisagisticas.

7-oferta do territorio para o recreio.

8a-acessibilidade do territorio.

Posso abandonar o metaconhecimento, s/n?

n

O conhecimento sobre o dominio existe em REGRAS que trabalham

sobre valores ligados a varios CONCEITOS.
Para cada conceito conheço o numero e nome das suas instancias e as
ligacoes a outros conceitos.
Conforme o que desejar saber pode escrever uma das palavras
sublinhadas.Nao lhe interessando o metaconhecimento escreva NAO.

regras

A sua regra permitira calcular conceitos de que tipo?:

a)aptidao; f)actor; d)escriptor

a

Qual o numero de tal aptidao
se tiver duvidas chame meta)conhecimento.

9

aptidao9-novaaptidao

a9=0 se

f2=1

ou

f3\=0 e f3\=3

senao

a9=(30*f5+70*f2)/100

Posso abandonar o metaconhecimento,s/n?

n

O conhecimento sobre o dominio existe em REGRAS que trabalham
sobre valores ligados a varios CONCEITOS.
Para cada conceito conheço o numero e nome das suas instancias e as
ligacoes a outros conceitos.
Conforme o que desejar saber pode escrever uma das palavras
sublinhadas.Nao lhe interessando o metaconhecimento escreva NAO.

regras

A sua regra permitira calcular conceitos de que tipo?:

a)aptidao; f)actor; d)escriptor

f

Qual o numero de tal factor
se tiver duvidas chame meta)conhecimento.

8h

factor8h-novofactor

f8h=s se

d6=0

ou

d4=1

senao

f8h=0 se

d1\=0

senao

f8h=(20*d2+60*d3+2)/100

Posso abandonar o metaconhecimento,s/n?

s

Quer entrar no modulo de Aquisicao, sim ou nao?

sim

s

qual e' a palavra chave?eco

AQUISICAO DE REGRAS

A regra a modificar e' ja' existente ou e' nova?

n

A sua regra permitira calcular conceitos de que tipo?:

a)ptidao; f)actor

a

Qual o numero de tal ptidao
se tiver duvidas chame meta)conhecimento.

0

Nome da regra?

outraptidao

Use a seguinte sintaxe:

<REG> ::= <ALT> senao <COR> | <ALT> | <COR>

<ALT> ::= <IGU> se <CND> | <IGU>

<CND> ::= <DIJ> | <CJC> | <ALT>

<IGU> ::= <CON> = <VAL> | <CON> = <CON>

<COR> ::= <EXP> | <ALT> | <REGRA>

<CON> ::= <letra minuscula> <ALF>

<DIJ> ::= <CJC> ou <CJC> | <CJC>

<ALF> ::= <numero> | <numero> <letra minuscula>

<CJC> ::= <IGU> e <IGU> | <IGU>

<VAL> ::= <numero> | <numero> <letra minuscula>

<EXP> ::= <CON> = <PAR> + <PAR> | <CON> = <PAR> | <REGRA>

<PAR> ::= <numero> * <CON> | <numero> | <CON>

Escreva a regra com letras minusculas. Nomeie o valor desta regra por a0
Termine por ponto (.)

a0=0 se f2a=0 ou f8a=s e f1=2 ou f4=3 senao a0=a se f9=1 e f6a=0 senao a0=

posso memorizar a nova regra? sim ou nao?

sim

Mais alteracoes s/n?

n

quer ver a regra? s/n

s

aptidao0-outraptidao

a0=0 se

f2a=0

ou

f8a=s e f1=2

ou

f4=3

senao

a0=a se

f9=1 e f6a=0

senao

a0=(1+f2+40*f3)/100

Com vista 'a actualizacao do metaconhecimento diga se as modificacoes
sao definitivas. Escreva s)im ou n)ao.

n

Quer consultar o metaconhecimento? s/n

n

adeus

DICIONÁRIO NUCLEAR

/* ##### [DIN] ##### */

?-op(30,xfy,-).
?-op(35,xfx,!).

/* ----- DICCIONARIO NUCLEAR ----- */

adeus(adeus).

em(F(em)).
de(F(de)).
com(F(com)).
sem(F(sem)).
para(F(para)).
entre(F(entre)).

ao(conc(F(a)+d(c:os,m-s,def-pos))).
aos(conc(F(a)+d(c:os,m-p,def-pos))).

do(conc(F(de)+d(c:o ,m-s,def-pos))).
dos(conc(F(de)+d(c:os,m-p,def-pos))).
da(conc(F(de)+d(c:a ,f-s,def-pos))).
das(conc(F(de)+d(c:as,f-p,def-pos))).

dum(conc(F(de)+d(um:um ,m-s,indf-pos))).
duns(conc(F(de)+d(um:uns ,m-p,indf-pos))).
duma(conc(F(de)+d(um:uma ,f-s,indf-pos))).
dumas(conc(F(de)+d(um:umas,f-p,indf-pos))).

no(conc(F(en)+d(o: ,m-s,def-pos))).
nos(conc(F(en)+d(o:os,m-p,def-pos))).
na(conc(F(en)+d(o:a ,f-s,def-pos))).
nas(conc(F(en)+d(o:as,f-p,def-pos))).

num(conc(F(em)+d(um:um ,m-s,indf-pos))).
nuns(conc(F(em)+d(um:uns ,m-p,indf-pos))).
numa(conc(F(em)+d(um:uma ,f-s,indf-pos))).
numas(conc(F(em)+d(um:umas,f-p,indf-pos))).

cujo(conc(cujo+d(o: ,m-s,def-pos))).
cujos(conc(cujo+d(c:os,m-p,def-pos))).
cuja(conc(cujo+d(o:a ,f-s,def-pos))).
cujas(conc(cujo+d(o:as,f-p,def-pos))).

onde(onde).

ou(ou).

e(e).
mas(e).

esta(M) :- 'e''(M).

'e''(v(ser:acento(e),L-1,adj)).
sao(v(ser:sao ,L-1,adj)).

seja(v(ser:seja,L-1,adj)).
sejam(v(ser:sejam,L-1,adj)).

ha(v(existir:acento(ha),L,L)).
existe(v(existir:existe ,L-s,L)).
existem(v(existir:existem,L-p,L)).

'tem'(v(ter:tem,L-p,obj)).
tem(v(ter:tem ,L ,obj)).

```

tenham( v(ter:tenham,L-P,obj) ).

tendo( tendo ).

vale( v(valer:vale ,L-s,obj) ).
valem( v(valer:valem,L-P,obj) ).

sera( sera ).

'da-me'( da-me ).
'de-me'( da-me ).

'diz-me'( 'diz-me' ).
'disa-me'( 'diz-me' ).

nao( nao ).

que( que+P(que) ).

qual( qual(qual:qual ,L-s) ).
quais( qual(qual:quais,L-P) ).

quanto( qual(quanto:quanto ,m-s) ).
quanta( qual(quanto:quanta ,f-s) ).
quantos( conc((quanto:quantos)+(m-P)) ).
quantas( conc((quanto:quantas)+(f-P)) ),

o( d(o:o ,m-s,def-pos) ).
os( d(o:os,m-P,def-pos) ).

/* 'a' esta' no DIA.PL */

as( d(o:as,f-P,def-pos) ).

um( d(um:um ,m-s,indf-pos) ).
uns( d(um:uns ,m-P,indf-pos) ).
uma( d(um:uma ,f-s,indf-pos) ).
umas( d(um:umas,f-P,indf-pos) ).

algun( d(algun:algun ,m-s,indf-pos) ).
alguns( d(algun:alguns ,m-P,indf-pos) ).
algunha( d(algun:algunha ,f-s,indf-pos) ).
algunhas( d(algun:algunhas,f-P,indf-pos) ).

cada( d(cada:cada,L-s,def-pos) ).

todo( d(todo:todo ,m-s,def-pos) ).
todos( d(todo:todos,m-P,def-pos) ).
toda( d(todo:toda ,f-s,def-pos) ).
todas( d(todo:todas,f-P,def-pos) ).

qualquer( d(qualquer:qualquer ,L-s,indf-pos) ).
quaisquer( d(qualquer:quaisquer,L-P,indf-pos) ).

nenhum( d(nenhum:nenhum ,m-s,indf-neg) ).
nenhuma( d(nenhum:nenhuma ,f-s,indf-neg) ).
 nenhuns( d(nenhum: nenhuns ,m-P,indf-neg) ).
nenhumas( d(nenhum:nenhumas,f-P,indf-neg) ).

mesmo( adj(mesmo:mesmo,m-s,obj) ).
mesma( adj(mesmo:mesma,f-s,obj) ).

muito( muito ).

```

/* ===== */
/* ===== */

DICIONÁRIO ESPECIFICO

/* **** [DIA] **** */

?-op(30,xfy,-).

?-op(35,xfx,;).

/* ----- DICCIONARIO VARIABEL (ORBI) ----- */

a(d(c:a, f-s, def-pos) + (F(a) + nca(artidao:artidao-L,f-s))).

maximo(M) :- maior(M).

maxima(M) :- maior(M).

minimo(M) :- menor(M).

minima(M) :- menor(M).

menos(ADJ(menor:menos, L,F(que))).

mais(ADJ(maior:mais, L,F(que))).

superior(ADJ(maior:superior, L-s,F(a))).

superiores(ADJ(maior:superiores, L-P,F(a))).

inferior(ADJ(menor:inferior, L-s,F(a))).

inferiores(ADJ(menor:inferiores, L-P,F(a))).

isual(ADJ(igual:isual, L-s,F(a))).

isuais(ADJ(igual:isuais, L-P,F(a))).

diferente(ADJ(diferente:diferente, L-s, F(de))).

diferentes(ADJ(diferente:diferentes, L-P, F(de))).

menor(ADJ(menor:menor, L-s,F(que))).

menores(ADJ(menor:menores, L-P,F(que))).

maior(ADJ(maior:maior, L-s,F(que))).

maiores(ADJ(maior:maiores, L-P,F(que))).

apto(apto).

apta(apto).

pouco(pouco).

muito(muito).

b(b).

c(c).

s(s).

h(h).

i(i).

J(j).

k(k).

valor(NC) :- nouns(valor,valor,NC).

valores(NC) :- nouns(valor,valores,NC).

media(NC) :- nouns(media,media,NC).

medias(NC) :- nouns(media,medias,NC).

resiao(NC) :- nouns(resiao,resiao,NC).

resioes(NC) :- nouns(resiao,resioes,NC).

rs(nca(resiao:resiao -L,f-s)).

Ponto(NC) :- nouns(ponto,ponto,NC).

Pontos(NC) :- nouns(ponto,pontos,NC).

P(nca(ponto:ponto -L,rms)).

artidao(NC) :- nouns(artidao,artidao,NC).

artidoes(NC) :- nouns(artidao,artidoes,NC).


```
factor(NC) :- ncms(factor,factor,NC).
factores(NC) :- ncmp(factor,factores,NC).
f(      nca(factor:factor  _ ,m-s) ).

descriptor(NC) :- ncms(descriptor,descriptor,NC).
descriptores(NC) :- ncmp(descriptor,descriptores,NC).
d(      nca(descriptor:descriptor  _ ,m-s) ).

representatividade(NC) :- ncfs(representatividade,representatividade,NC).
representatividades(NC) :- ncfr(representatividade,representatividades,NC).
r(      nc(representatividade:representatividade  _ ,f-_) ).

resta(NC) :- ncfs(resta,resta,NC).
restas(NC) :- ncfr(resta,restas,NC).
```

```
ncms(T,E,nc(T;E-_,m-s)).
ncmp(T,E,nc(T;E-_,m-p)).
ncfs(T,E,nc(T;E-_,f-s)).
ncfr(T,E,nc(T;E-_,f-p)).
```

```
/* ===== */
/* ===== */
```

MORFOLOGIA

```
/* #### [ MOA.PL ] #### */
```

```
/* ----- MORFOLOGIA -----
```

```
/*----- morfologia, e lancamento da gramatica -----
```

```
go :- repeat,  
    try(retract(explic(_))),  
    try((retract(imprime),tell('!f:'),nl,told)),  
    try(retract(P_(_))),  
    write('- '),  
    morfologia(LCM,LI),  
    (LI=[],  
     LCM\=[]),  
    nl,  
    tell('mg:B'), write(cont(LCM)), put(46), told, !, LCM=[adeus], halt,  
    LI\=[], saida(mess(s1)), write(LI), nl,nl, fail ).
```

```
morfologia(NM,D) :- frase(Ps), lf((wr(Ps),nl,nl)),  
    morf(Ps,M,D), ilo(M,NM), !.
```

```
ilo(['diz-me'ID],D),  
ilo(D,D).
```

```
/* wr no ORBI.PL */
```

```
wri(n(S)) :- write(S).  
wri(virs) :- put(44).  
wri(pontoevirs) :- put(59).  
wri(S) :- write(S).
```

```
frase(Ps) :- get(C), palavras(C,Ps), !.
```

```
palavras(C,[]) :- terminador(C), term(C), !.
```

```
palavras(C,Ps) :- espacador(C), !, frase(Ps).
```

```
palavras(C,[F|Ps]) :- inteiro(C,O,F,NC), palavras(NC,Ps), !.
```

```
palavras(34,[meta(LP)|Ps]) :- !, set(C), meta(C,LP), !, cont(LP), frase(Ps).
```

```
palavras(C,[F|Ps]) :- separador(C,F), !, frase(Ps).
```

```
palavras(C,[F|Ps]) :- letra(C,LC),  
    palavra(LC,NC,CP), name(F,CP), !, palavras(NC,Ps).
```

```
term(46) :- skip(10).
```

```
term(10).
```

```
term(42) :- assert(imprime), set0(C), term(C),  
    tell('mg:B'), write(imp), put(46), told.
```

```
term(43) :- set0(C), ( C=43, D='s', set0(E) ; D='n', E=C ),  
    assert(explic('s'/D)),  
    term(E).
```

```
term(_ ) :- set0(C), term(C).
```

```
terminador(63), terminador(46), terminador(33).
```

```

terminador(42).  terminador(43).

espacador(32).  espacador(63).  espacador(46).  espacador(33).  espacador(10).

separador(44,virs).  separador(59,pontcevirs).  separador(34,'').
separador(96,acento_grave).

palavra(C,C,[]) :- espacador(C) ; separador(C,_); terminador(C).
palavra(39,NC,LC) :- set0(C1), palavra(C1,NC,LC).
palavra(101,NC,[101|Cn]) :- set0(C2),
    ( C2=39, NC=32, Cn=[39] ; palavra(C2,NC,Cn) ).
palavra(C1,NC,[C1|Cn]) :- set0(C2),
    (not (C2=<57, 48=<C2), palavra(C2,NC,Cn) ;
     NC=C2, Cn=[] /* inteiro */ ).

letra(C,LC) :- C=<90, 65=<C, LC is C+32.
letra(C,C).

inteiro(C,N,I,NC) :- C=<57, 48=<C, NN is N*10+C-48, set0(PC),
    ( inteiro(PC,NN,I,NC) ; NC=PC, I=n(NN) ).

meta(34,[]) :- !.
meta(C,_ ) :- terminador(C), nl, saida(mess(s4)), !, fail.
meta(C,[P|Ps]) :- palavra(C,NC,LC), name(P,LC), !,
    ( branco(NC), !, set(NNC), meta(NNC,Ps) ;
      meta(NC,Ps) ).
branco(32).

comp([_] ) :- !.
comp(_ ) :- nl, saida(mess(s3)), nl,nl, fail.

morf([],[],[]).
morf([F1|Fn],[M1|M],I) :- ( dic(F1,M1) ; separador(_,F1), M1=F1 ),
    morf(Fn,M,I).
morf([F1|Fn],M,[F1|D]) :- morf(Fn,M,I).

dic(meta(M),meta(M)).
dic(n(N),n(N)).
dic(F,M) :- functor(F,F,1), arg(1,D,M), !.

```

```

/* =====>
/* =====>

```

SINTAXE NUCLEAR

```
/* #### [ SINN.FL ] #### */
```

```
?-op(30,xfu,-).  
?-op(35,xfk,!).
```

```
/* ===== GRAMATICA NUCLEAR ===== */
```

```
solve(G) :- G, !.
```

```
/* ===== PERGUNTAS e ORDEM ===== */
```

```
frase(LE,que(MF,FF),LS) :- /* (da'-me/o que e') + sn */  
  extrap(LE,nc,L1,EX,EXS),  
  ( opc(L1,L2),  
    extrap(L2,nc,L3,EX,EXS) ;  
    L1=L3 ),  
  sn(L3,pos:D,AN,MN,FI,_,E,EX,SN,LS),  
  supsn(E,FI,F),  
  testcada(F,FFI,MN,MFI),  
  semextrap(EXS,FFI,FF,MFI,MF),  
  assert_elipse(elipse(que,EXS,[J]+SN)).
```

```
frase(LE,que(MF,FF),LS) :- /* qual/quais + (e'/sao) + sn */  
  extrap(LE,c,[qual(QU,AQ)]IL1,EX,EXS), /* OU quanto + vale/valem + sn */  
  ( ( QU=qual:_,  
    L1=[v(ser:T1,AV,_)IL2], !,  
    acordo(AQ,AV,QU,ser:T1) ;  
    QU=quanto:_, !,  
    AQ=_s,  
    L1=[v(valer:T2,AV,_)IL2] ),  
  extrap(L2,nc,L3,EX,EXS) ;  
  L1=L3 ),  
  sn(L3,_,AN,MN,FI,_,E,EX,SN,LS),  
  ( QU=qual:_,  
    acordo(AQ,AN,QU,MN) ;  
    QU=quanto:_,  
    acordo(AV,AN,valer:T2,MN) ),  
  segue1(E,FI,MN,EXS,FF,MF,SN),  
  segue1(E,FI,MN,EXS,FF,MF,SN) :-  
  supsn(E,FI,F),  
  testcada(F,FFI,MN,MFI),  
  semextrap(EXS,FFI,FF,MFI,MF),  
  assert_elipse(elipse(que,EXS,[qual(qual:_,_)IL1]+SN)).
```

```
frase(LE,que(MF,NSF),LS) :- /* (pref) + que/qual + sn + (e'/tole) + pto  
  ( extrap(LE,g(TC,gueta(que)),L1,EX,EXS), pref(TC,L1,[gueta(que)]IL2),  
    extrap(LE,g(TC,qual(QU,AQ)),L1,EX,EXS), pref(TC,L1,[qual(QU,AQ)]IL2),  
    sn([det]IL2,_,AS,MS,SS,ST,_,EX,[det]LND+LCOMPN+RELATS,L3),  
    acordo(AQ,AS,QU,MS), se_ome+L3,L4),  
  ( TC=obj,  
    props(suj(AS):dn(pos:_,MS,SS,ST,RELATS),SF,EX,L4,L3) ;  
    props(comp(TC):dn(pos:_,MS,SS,ST,RELATS),SF,EX,L4,L3) ),  
  segue2(EXS,SF,NSF,MS,MF,L2,L1,SN,LCOMPN,RELATS,L4).
```

```
segue2(EXS,E:(S,SU),NSF,MS,MF,L2,L1,SN,LCOMPN,RELATS,L4) :-  
  aus(E,S,SU,SF),  
  conc(PREFQU,L2,L1),  
  testcada(SF,NSFI,MS,MFI),
```

```
semextrap(EXS, NSPI, NSF, MFI, MF),
assert_ellipse(ellipse(que, EXS, PREPQU+(SN+LCOMPN+RELATS)+L4)).
```

```
frase(LE, quant(R, F), LS) :-
    /* (PREP) + quantos/quantas + sn + (e' + que + prof) */
    extrap(LE, q(TC, conc((quanto:T)+(G-F))), L1, EX, EXS),
    prep(TC, L1, [conc((quanto:T)+(G-F))|L2]),
    sn([d(o:_, (G-F), _) | L2], pos:_, A, M, S,
        v(ST, (al_(nil, (ST, REL, SV), C), length(C, N))), _, EX, SN, L3),
    acordo((G-F), A, quanto:T, M),
    sesue3(A:R, N, EX:EXS, F, LS, TC, dn(pos:_, M, S, ST, RELATS),
        RELATS, SN, SV, L2, L1, L3),
    sesue3(A:R, N, EX:EXS, F, LS, TC, DN, RELATS, [L1SN]+LCOMPN+RELATS, SV, L2, L1, L3),
    e_que(L3, L4),
    ( TC=obv,
        props(suj(A):DN, E:(SS, SV), EX, L4, LS), T=true ;
        props(comp(TC):DN, E:(SS, T), EX, L4, LS), SV=true ),
    aus(E, SS, T, FI),
    semextrquant(EXS, N, FI, F, R),
    conc(PREPQU, L2, L1),
    assert_ellipse(ellipse(quant, EXS, PREPQU+(SN+LCOMPN+RELATS)+L4)).
```

```
frase(LE, quant(R, FF), LS) :-
    /* quantos/quantas + (sao) + sn */
    extrap(LE, c, [conc((quanto:T)+AQ) | L1], EX, EXS),
    ( L1=[v(ser:T1, AV, _) | L2] ; L2=L1 ),
    extrap(L2, nc, L3, EX, EXS),
    sn(L3, pos:_, AN, MN, F,
        v(ST, (al_(nil, (ST, REL, true), C), length(C, N))), _, EX, SN, L5),
    acordo(AQ, AV, quanto:T, ser-T1),
    acordo(AQ, AN, quanto:T, MN),
    semextrquant(EXS, N, F, FF, R),
    assert_ellipse(ellipse(quant, EXS, [conc((quanto:L)+L), v(ser:L, L, L), SN])).
```

```
frase(LE, simnao(L, NSF), LS) :-
    /* (sera' + que) + sn + prof x,
    /* (sera' + que) + (nao) + existir + sn */
    extrap(LE, nc, L1, EX, EXS),
    ( L1=[sera, que+F(que) | L2],
        extrap(L2, nc, L3, EX, EXS) ;
        L1=L3 ),
    ( L3=[nao | L31], FV=nes ; L3=L31, FV=pos ),
    L31=[v(existir:T, AE, L3) | L4],
    extrap(L4, nc, L5, EX, EXS) ;
    L5=L3 ),
    det(L, L5, L6),
    sesue4(FV, L5, AE, L3, L6, EX, EXS, NSF, LS),
    sesue4(FV, L5, AE, L3, L6, EX, EXS, NSF, LS) :-
    sn(L6, FI, AS, MS, SS, STS, E, EX, SN+LCOMPN+RELATS, L7),
    acordo(AE, AS, existir:T, MS),
    ( L3\L5,
        duplane(FI, FI, NFV),
        L7=LS, formaLV(NFV, (SS, true), SP) ;
        props(suj(A):dn(FI, MS, SS, STS, RELATS), E:(S, SV), EX, L7, LS),
        aus(E, S, SV, SP) ),
    atorceia(SP, NSF),
    assert_ellipse(ellipse(simnao, EXS, [L]+(SN+LCOMPN+RELATS)+L7)).
```

```
aque([v(ser:L, L, L), que+F(que) | L], L) :- !, /* e' + que */
aque(L, L).
```

```
afc([da-me | L], L).
```

opc([d(_,_),quet_,v(ser:_,_)]L),L).


```

/* ===== ELIPSES =====
  <X> : elipse total de X OU [X] : elipse parcial de X
)

elipse(FQ,PREPQU+(SNB+LCOMPEN+RELATS)+PROP,Q,LE,LS) :-
( LCOMPEN=[], functor(Q,FQ,2),
  prep=(Pref),LE,L1), Pref\=com,
  encontrar_snb(L1,MN1),
  encontrar_prep(L,LCOMPEN,MN2,[],LELIF,FIMCOMP),
  conclist([PREPQU,SNB,LELIF,LE],LY),

( MN1\=MN2,
  conc(LY,PROP,LZ) /* <(Pref)+que+snb> + compen + relats + <prop>
                  /* OU <(Pref)+quantos+snb> + compen + relats + <prop>
                  /* OU <(sera+que)+snb> + compen + relats + <prop>
  ;
  MN1=MN2,
  conclist([LY,FIMCOMP,RELATS,PROP],LZ)
  /* <(Pref)+que+snb>+[compen]+<relats+prop>
  /* OU <(Pref)+quantos+snb>+[compen]+<relats+prop>
  /* OU <(sera+que)+snb>+[compen]+<relats+prop>
)

conclist(LCOMPEN,COMPNS),

( functor(Q,FQ,2), solve(( F=[ ] ; F=PREPQU )),
  ( conclist([F,LE,COMPNS,RELATS,PROP],LY)
    /* (Pref) + que/<> + snb + <compen+relats+prop>
    /* OU (Pref)+quantos/<>+ snb + <compen+relats+prop>
    /* OU (sera+que) + snb + <compen+relats+prop>
  ;
  conclist([F,LE,PROP],LY) /* (Pref) + que + sn + <prop>
  /* OU (Pref) + quantos + sn + <prop>
  /* OU (sera+que) + sn + <prop>
  ;
  conclist([F,SNB,COMPNS,RELATS,LE],LY)
    /* <(sera+que)+sn>/<QU+sn>/<sn> + prop
  ), LZ=LY
)

; PREPQU=[], /* muda so/ o quantificador
  conclist([LE,SNB,COMPNS,RELATS,PROP],LZ)
)

), elfrase(LZ,Q,LS),

```

```

elipse(FQ,QU+(SNB+LCOMPEN+RELATS),Q,LE,LS) :- FQ\=simnac,

```

```

( LCOMPEN=[], functor(Q,FQ,2),
  prep=(Pref),LE,L1), Pref\=com,
  encontrar_snb(L1,MN1),
  encontrar_prep(L,LCOMPEN,MN2,[],LELIF,FIMCOMP),
  conclist([QU,SNB,LELIF,LE],LR),

( MN1\=MN2, LZ=LR
  /* <qual/quantos+snb> + compen + relats
  /* OU <quantos+snb> + compen + relats
  /* OU <dan+snb> + compen + relats
  ;
  MN1=MN2,
  conclist([LZ,FIMCOMP,RELATS],LR)
  /* <qual/quantos+snb> + [compen] + relats
)

```

```

/* OU <quantos+saos+snb> + [comen.] + relats >
/* OU <dar+me+snb> + [comen.] + relats >
)
;

concllist(LCOMPEN,COMPNS),

( functor(Q,FQ,2),
( solve(( F=I] ; F=QU ))),

concllist([F,LE,COMPNS,RELATS],LZ)
/* qual/quantos+vale/<> + snb + <comen+relats> >
/* quantos + saos/<> + snb + <comen+relats> >
;
/* dar + me + snb + <comen+relats> >

conc(QU,LE,LZ)
/* <qual/quantos+vale> + snb >
/* <quantos+saos> + snb >
/* <dar+me> + snb >
;
concllist([QU,SNB,COMPNS,LE],LZ) /* QU + <snb+comen> + relats >
;

F=I], /* <qual/quantos+vale> + <snb + comen> + relats */
concllist([LE,SNB,COMPNS,RELATS],LZ)
) ;

QU\=I], /* muda so' o quantificador >
concllist([LE,SNB,COMPNS,RELATS],LZ)

)

), elfrase(LZ,R,LS).

```

```

elfrase(L1,R,L2) :- Juntar(acordo), frase(L1,R,L2).

```

```

encontrar_Lsn(LE,NOME) :- det(L,LE,NLE),
sn(NLE,_,_,NOME;_,_,_,_,_,_,_), !.

```

```

encontrar_Larea(F,[COMPEN|LCOMPEN],NOME,ANTIGA,ELIPSE,FIMCOMP) :-
area(F(F),COMPEN,L1),
encontrar_Lsn(L1,NOME),
ELIPSE=ANTIGA,
transformar(LCOMPEN,FIMCOMP) ;
conc(ANTIGA,COMPEN,NOVA),
encontrar_Larea(F,LCOMPEN,NOME,NOVA,ELIPSE,FIMCOMP).

```

```

transformar([],[]).

```

```

transformar([X|[]],X).

```

```

transformar([X|Y|Z],W) :- transformar(Z,_,conc(X,Y,A),conc(A,F,W)).

```

```

concllist([],[]).

```

```

concllist([L],L).

```

```

concllist([L1,L2|SL],L) :- concllist([L2|SL],L3), conc(L1,L3,L).

```

```

/* ===== EXTRAPOSICAO ===== >

```

```

extrap(LE,C,[FILS],EX,_) :-
  var(EX),
  solve( ( ( C==c, member(P,LE) ; C==nc ; C=g(_,Q), member(Q,LE) ),
    vrs(LE,L0),
    ( prep(P(Prep),L0,L1), Prep\=a ; C=g(P(_),Q), L1=L0 ),
    L1=[[]],
    in(d(DE:_,_,_),D),
    sn(L1,_,ME,_,SE,_,_,L2),
    conc(E:L2,L0),
    vrs(L2:[FILS]),
    asserta(extrap(EX,[FILS],s(DE,ME,SE))) ) ),
  fail.

```

```

extrap(,_,LS,EX,EXS) :- retract(extrap(EX,LS,EXS)), !.

```

```

extrap(L,_,L,_,_).

```

```

vrs([vrs|L],L).

```

```

vrs( L ,L).

```

```
extrap(LE,C,[FILS],EX,_) :-
  var(EX),
  solve( ( ( C=c, member(P,LE) ; C=nc ; C=g(,Q), member(Q,LE) ),
    vrs(LE,L0),
    ( Pref(F(Pref),L0,L1), Pref\=a ; C=g(F(,),Q), L1=L0 ),
    L1=[],
    in(d(DE:,ME:,SE:),D),
    sn(L1,ME,SE,L2),
    conc(E:L2,L0),
    vrs(L2:[FILS]),
    asserta(extrap(EX,[FILS],s(DE,ME,SE))) ),
  fail.
```

```
extrap(,_,LS,EX,EXS) :- retract(extrap(EX,LS,EXS)), !.
```

```
extrap(L,_,L,_,_).
```

```
vrs([vrs|L],L).
```

```
vrs( L ,L).
```

```

/* ===== VERBOS, NEGACAO, ADJECTIVOS e ARGUMENTOS ===== */

sv([nao|L1],FD:_,AV,VE,MV,ADJ,TC,FV,EX,LS) :- /* nao + v + (adv) */
  nv(L1,AV,VE,MV,ADJ,TC,EX,L2),
  doublenes(FD,L2,FV,L3), !,
  det(MV,L3,LS).

sv(LE,FD:_,AV,VE,MV,ADJ,TC,FV,EX,LS) :- /* v + (adv) */
  nv(LE,AV,VE,MV,ADJ,TC,EX,L1),
  ( FD=pos; FV=pos, L1=L2 ;
    doublenes(pos,L1,FV,L2) ), !,
  det(MV,L2,LS).

det(_,L,L).
det(ter:_,L,[det|L]). /* ter -> ter + det */

/*****
det(ter:_,[P,pronom(MP,AP)|L],[B|L]) :- /* ter o seu -> ter */
  /* in(d(MD,AD,def-pos),B),
  acordo(AD,AP,MD,MP).
*****/

nv([V(M:T1,AV,P)|L],AV,VE,MV,ADJ,TC,_,LS) :-
  ( M=existir; MV=M:T1, LS=LE, TC=P ; /* existir / haver / ter */
    M=ter,
    solve(( ( after(LE,LS,MV) ; ADV==adv, LS=LE, MV=M:T1, TC=P ) ) ) ;
  ( ( M=ser, opm(LE,L1,MA) ; M=valer, L1=LE ) ;
    M=ter, LE=[inc(valor:_,_)|L1] ), /* valer / ser + adv */
  advj(L1,MA:T2,AA,TC,VB,LS),
  acordo(AV,AA,M:T1,MA:T2),
  MV=VB:MA. /* advj = _ em ; ser + entre .. */
/* em ; ser + 1/2/ato/.. */

after(LE,LS,(ser(V):isual)) :- ( LE=[d(.,.,_)|L1] ; L1=LE ),
  L1=[inc(valor:_,_)|H],
  sn_valor(H,LS,V).

opm([d(c:.,.,_)|L],L,MA) :- member(MA,[maior,menor]),
  opm(L,L,_).

advj([r(entre)|L],entre:L,_,obj,ser(L),[r(entre)|L]) :- !, /*entre -> entre */
advj([adv(MA:A,AA,P)|L],MA:A,AA,TC,ser(L),LS) :-
  ( TC=P ; member(MA,[maior,menor]) ), !, /* isual/inferior/menor.. */
advj(LE,isual:L,_,obj,ter,LE) :-
  sn_valor(LE,L,_), /* em ; ser/valer + 1/2/ato/.. */

duplanes(.,es,pos:L,nes),
duplanes(.,.,.,pos).

doublenes(pos,[d(M:_,A,indf-_)|L],nes,[d(ur:_,A,indf-pos)|L]) :-
  M=nenhum ; M=qualquer.
doublenes(pos,L,nes,L).
doublenes(nes,L,pos,L).

fsJcF(ESUJ,ECOMP) :- mesF(''), write(ESUJ),

```

```
write(' nao pode ser sujeito do verbo 'ter' '),
write('com o complemento '), write(ECOMP), mesf('.'.').
```

```
/* ===== PREPOSICAO ===== */
```

```
pref(P(X),[CILS],[DILS]) :-
  in(conc(P(X)+I),C), !.
```

```
pref(P(X),[FILS],LS) :-
  in(P(X),P), !.
```

```
pref(obj,L,L).
```

```
/* ===== PROPOSICOES ===== */
```

```
/* ---- conjuncoes de proposicoes ---- */
```

```
props(AUS,E:(NSS,SV),EX,LE,LS) :- prop(AUS,E1:(NSS1,SV1),EX,LE,L1),
  ( conc(X,L1,L2),
    elisao(L2,LE,L3),
    eli_suj_naus(AUS,NSS1,NAUS,LE,L3,L4),
    props(NAUS,En:(NSS,SVn),EX,L4,LS),
    ge(E1-En,E), conc(X,SV1,SVn,SV) ;
    L1=LS, E=E1, NSS=NSS1, SV=SV1 ).
```

```
conj(e,A,B,(A,B)).
conj(ou,A,B,(A;B)).
```

```
elisao([VIR],_,[VIR]) :- ( V=v(L,L,L) ; V=nao ), !.
```

```
elisao([A1IL1],[V,A2IR2],L3) :- eli(A1,[A1IL1],V,A2,R2,L3).
```

```
elisao(L2,[VIL1],[VIL2]).
```

```
eli(adj(L,L,L),L,L,L,[V(ser;L,L,L)IL3]).
```

```
eli(n(L),L,V,L,L,[VIL3]).
```

```
eli(P(L),L,V,A2,L,[V,A2IL3]).
```

```
eli(L,L,V,A2,[P(P)IL3],[V,A2,P(P)IL3]).
```

```
eli(L,L,V,A2,L,[V,A2,L]).
```

```
eli_suj_naus(AUS:dn(FD,M,S,ST,R), NS, AUS:dn(FD,M,NS,ST,R), LE, L3, L4) :-
  AUS=suj(L), L4=L3 ;
  AUS=comp(L),
  sn(LE,L,L,L,L,L,dn(FD,M,NS,ST,R),L,L),
  conc(SN,L,LE),
  conc(SN,L3,L4).
```

*) ===== SINTAGMAS NOMINAIS ===== *

/* ---- SN = snb + (compn) + (relats) ---- */

```

n(LE,FD,AN,MN,NSD,ST,E,EX,(SNB+LCOMPN+RELATS),LS) :-
  snb(LE,FD,AN,MN,ST,S1,F1,VE,L1),
  compn(MN,ST,_,true,F1,VES,EX,LCOMPN,L1,L2),
  ae(VE-VES,E),
  cadadecada(SI,[],SI),
  relats(L2,suJ(AN):dn(FD,MN,SI,ST,[]),NSD,EX,sn,LS),
  conc(RELATS,LS,L2),
  conc(SNB,L1,LE).          /* relats, snb : para elipse */

```

/* ---- SN = snb + relats + compn ---- */

```

n(LE,FD,AN,MN,NSD,ST,E,EX,(SNB+LCOMPN+RELATS),LS) :-
  snb(LE,FD,AN,MN,ST,SD,F1,VE,L1),
  nome(MN,ST),
  relats(L1,suJ(AN):dn(FD,MN,SI,ST,[]),NSD,EX,sn,L2), L1\=L2,
  compn(MN,ST,_,true,F1,VES,EX,LCOMPN,L2,LS), !,
  ae(VE-VES,E),
  cadadecada(SD,[],SI),
  conc(RELATS,L2,L1),
  conc(SNB,L1,LE).          /* relats, snb : para elipse */

```

/* ---- sintagma de base (SNB) ---- */

* A primeira clausula de snb e' particular de aplicacao e nao esta' aqui *

```

n(Ed(D,AD,def-pos)[]LJ,Pos:cada,AN,MN,_,cada([MN],F),F,nil,LS) :-
  E=cada:L, try((L=Ed(D,_,_),conc(L),NILS) ; L=[NILS]) ;
  D=todo:L, L=[ED,NILS],
  in(d(M,A,def-pos),N),          /* cada + no */
  M=o:L, acordo(AD,A,D,M), !, /* DU todo(s)/toda(s) + o(s)/a(s) + no */
  in(nc(MN,AN),N),
  acordo(A,AN,M,MN).

```

```

n(Ed(qualquer:T,AD,indf-pos),
  NILS],Pos:cada,AN,MN,_,cada([MN],F),F,nil,LS) :-
  in(nc(MN,AN),N), !,
  acordo(AD,AN,qualquer:T,MN).          /* qualquer/quaisquer + no

```

```

n(Ed(MD,AD,indf-FD),NILS],F1:I,AN,MN,_,NF,F,nil,LS) :-
  (FD=Pos, L=[Ed(qualquer:T,A,_)[]LJ], acordo(AD,A,MI,qualquer:T) ; LS=L),
  in(nc(MN,AN),N),
  acordo(AD,AN,MD,MR),
  (FD=Pos, I=um, NF=F ;          /* um(ns)/uma(s)/algun(ns)/alguna(s) */
  D=cada, NF=cada([MN],F)), !.      /* nenhum(ns)/nenhuma(s) + no */

```

```

n([n(N),X[]LJ],Pos:cada,AN,MN,v(ST,(all(MN,(ST,REL,_)S),length(S,N))),F,
  nil,LS) :-
  in(nc(MN,AN),X),
  (N>1, Q=F ; Q=s),          /* 0/1/2/3/.. + no */
  acordo(L-Q,AN,N,MN).

```

```

n(LE,Pos:o,AN,MN,_,F,F,VE,LS) :-
  artiso(LE,MA,AA,[adJ(SUP:L,AS,_)[]LJ],NILJ),
  acordo(AA,AS,MA,SUP),
  ( in(nc(MN,AN),N), gopt(L,MN,LS) ; /* mesmo(s)/mesma(s)/maior/menor + no */
  in(nca(MN,AN),N), no([MN]LJ+LS) ),
  acordo(AM,AN,SUP,MN),

```

```

suf(SUF,MN,VE).

artiso([AILS],MA,AA,LS) :- in(d(MA,AA,-Pos),A), !.
artiso(L,_,_,L).

suf(mesmo,M,M) :- !.

suf(T,I:S,T+S) :- member(T,[maior,menor]), fval(I), !. /* maior/menor */

/* ===== COMPLEMENTO DO SUBSTANTIVO ===== */
-----
compn(NT,ST,X,AF,F,VE,EX,LCOMPN,L,LS) :- nonvar(EX),
    conc(EX,L,LE),
    compn(NT,ST,X,AF,F,VE,_,LCOMPN,LE,LS).

compn(NT,ST,NT-ST,AF,(NST,AF),nil,EX,CD,L,L) :- nome(NT,ST), semLV2(ST,NST).

```


/* ===== RELATIVAS =====

relats(LE,SUJ,NSS,EX,O,LS) :-
 sujrel(SUJ,SUJR),
 relat(LE,SUJR,E:(SSI,SV),EX,L1),
 eli_suj_naus(SUJR,SSI,NSUJR,_,_,_),
 crelat(C,NSUJR,E:SV,NSS,EX,O,L1,LS), !.

relats(L,_:dn(,_,SS,SBS,_)NSS,_,_,L) :- quant(SBS,true,_,SS,NSS).

sujrel(AUS:dn(FD:_,M,S,SB,R),AUS:dn(FD:rel,M,S,SB,R)).

crelat(C,SUJ,EI:SRI,NSS,EX,O,LE,LS) :-
 solve((conj(C,LE,L1) ; LE=LI)),
 relat(L1,SUJ,E:(SSI,SV),EX,L2),
 conj(C,SRI,SV,SR),
 eli_suj_naus(SUJ,SSI,NSUJ,_,_,_),
 crelat(C,NSUJ,EI-E:SR,NSS,EX,O,L2,LS).

crelat(,_:dn(,_,SS,SBS,_)E:SRI,NSS,_,O,L,L) :-
 ge(E,NE),
 (O=sn, quant(SBS,SRI,SR,SS,SSI) ;
 order(SS,SRI,SSI,SR)),
 aus(NE,SSI,SR,NSS), !.

/* -- QUE -- */

relat([que+P(que)IL1],SUJ:S,SR,EX,LS) :-
 PROPS(SUJ:S,SR,EX,L1,LS) ;
 PROPS(comp(obj):S,SR,EX,L1,LS).

/* -- PREP + QUE -- */

relat([P,que+P(que)IL1],_:S,SR,EX,LS) :-
 in(P(X),P),
 PROPS(comp(P(X)):S,SR,EX,L1,LS).

/* -- NAO TENDO = QUE NAO TEM -- */

relat([nac,tendoIL1],SUJ,SR,EX,LS) :-
 relat([O,nac,v(ter:_,_,obj)IL1],SUJ,SR,EX,LS).

/* -- COM (ou TENDO) ou SEM (ou NAO TENDO) -- */

relat([XILE1],SUJ,SR,EX,LS) :-
 ((X=P(com) ; X=tendo), L2=L1 ;
 X=P(sem), L2=[nacIL1]),
 L1=[v(ter:_,_,obj)ILE1],
 PROPS(SUJ,SR,EX,L2,LS).

/* -- elipse do relativo e do verbo 'ser' -- */

relat([ADJIL1],SUJ,SR,EX,LS) :-
 el_rel(ADJ),
 PROPS(SUJ,SR,EX,[v(ser:_,_,_)ADJIL1],LS).

el_rel(adj(,_,_)).

el_rel(adv(_)).

el_rel(P(entre)).

/* -- PREP + GUAL -- */

relat(LE,suj(AD):S,SR,EX,LS) :-
 contrac(P(P),LE,L1),
 L1=[D,gual(MQ,AQ)IL2],
 in(d(c:L,AD,def-_),D).

```
PROPS(COMP(F(F));S,SR,EX,L2,LS),
acordo(AC,AQ,MC,MQ),
acordo(AQ,AD,MQ,o:_).
```

```
/* ===== CONTRACCAO ===== x
```

```
contrac(X,[XIL],L) :- !.
contrac(X,[WILS],[YILS]) :- in(conc(X+Y),W).
```

```
/* ===== CONJUNCAO ===== x
```

```
conj(e,[EIL],L) :- E=e ; E=virg.
conj(ou,[ouIL],L).
```

```
/* ===== x
/* ===== x
```

SEMÁNTICA NUCLEAR

```
/* #### [ SEMN.PL ] #### */
```

```
/* ===== S E M A N T I C A ===== */
```

```
try(F) :- F, !.  
try(_).
```

```
/* ----- semantica da elipse com e sem. extraspecifica ----- */
```

```
assert_elipse(E) :- try(retract(elipse(_,_,_))), assertz(E).
```

```
mens :- mens(_), nl, retract(mens(0)), G, fail.
```

```
extraf_el(s(DE,_,_),Q,G) :- var(DE) ; Q=simneo(_).
```

```
extraf_el(EXS,que(M,F),que(NM,NF)) :- semextraf(EXS,F,NF,M,NM).
```

```
extraf_el(EXS,quant(R,all_(M,F,R)),quant(R,all_(NM,NF,R))) :-  
semextrquant(EXS,M,F,NF,NM).
```

```
/* ----- controle e semantica da : QU-.. CADA .. ----- */
```

```
testcada(cada(X,F), all_(V,F,S), MN, E=U) :-  
( MN=(media:_-V-_), E=media ;  
( not member(MN,X), conc(X,[MN],h) ; M=X ), evidenciam_(M,U,E) ).
```

```
testcada(all_(A,B,C), all_(U,B,C), _ , E=V ) :- evidenciam_(A,U,E).
```

```
testcada(sup(V,MV,S), S, M, E=V ) :- S=(all_([LV,MV],_),_),  
evidenciam_(M,LV,E), v(E,LV,MV).
```

```
testcada( F , F , MN, E=V ) :- evidenciam_(MN,U,E).
```

```
evidenciam_(M,U,E) :- evid(M,VI,EI), sem_lista(VI,U), sem_lista(EI,E).
```

```
evid([],[],[]).
```

```
evid([E:_-VILM],[VILV],[EILE]) :- evid(LM,LV,LE).
```

```
evid(N:E,[N,U],NE) :- var(N), evidenciam_(E,U,NE).
```

```
evid([E:_-V]:C,U:C,E).
```

```
evid(E:_-V,U,E).
```

```
sem_lista(Y,X) :- nonvar(Y), Y=[X].
```

```
sem_lista(X,X).
```

```
semextraf( s(DE,_,_), FF, FF, MF, MF ) :- DEN==cada.
```

```
semextraf( s(_:_:E-VE,SE), all_(M,E,C),  
all_( EP, (SE,al_(A,NE:C)), S ), EC1=_ , _=EP ) :-  
EP=(VE:(EC=C)),  
( M=[M1A], M1==VE, EC1=[_IEC] ;  
M=(X:(C_LIL3=CC)), A=(X:(L=CC)), EC=EC1 ;  
A=M, EC=EC1 ),  
novo(E,NE,VE).
```

```
novo(al_(M1A),C,E),al_(A,C,E),VE) :- VE==M, C=( _ same _ ),
```

```
novo( E, E, _ ).
```

```
semextraf( s(_:_:E-VE,SE), FF1, all_(VE:MF1,(SE,FF1), S ), MF1, E=(VE:FF1) ).
```

```

semextrquant( s(DE,_,_), _, all_(A,B,D), all_(U,B,D), [numero,E]=V) :-
    DE\==cada,
    evidencia_(A,U,E).

semextrquant( s(,_,:E-VE,SE), N, all_(A,B,D),
              all_(VE:N,(SE,al_(nil,B,D)),C), [VE,numero]=(VE:N)).

semextrquant( s(DE,_,_), N, F, NP, NN) :-
    sem_cada(F,NP), NP=(al_(A,_,_),_), DE\==cada, agr(A,N,NN).

semextrquant( _, N, cada(E_:E-VI_),F), all_(V:N,F,C), [E,numero]=(V:N)).

semextrquant( s(DE,_,_), N, F, all_(nil,F,C), N) :- DE\==cada.

semextrquant( s(,_,:E-VE,SE), N, F, all_(VE:N,(SE,F),C), [E,numero]=(VE:N)).

agr(,_,:E-):_,N,[numero,E]=N).

agr(,_,N,N).

sem_lv(v(S,P),S) :- nonvar(P), !.
sem_lv( S, S).

sem_lv2(v(,S),S) :- !.
sem_lv2(S,S).

forma_lv( Pos, (cada(,_(SC,F)),true), S ) :-!, forma_lv(Pos,(cada(,SC),F),S).
forma_lv( Pos, (cada(,SC),F), S ) :-!, not (SC,not F) ) :-!.
forma_lv( Pos, S, S ) :-!.
forma_lv( nes, (cada(,SC),F), S ) :-!, not (SC,F) ) :-!.
forma_lv( nes, S, S ) :-!, not S ) :-!.

/* ----- semantica da Proposicao -----*/

propcheia( cada(,_(SS,not (SC,not F))), not (SS,SC,not F) ) :-!.
propcheia( cada(,_(SS,S), not (SS,not S) ) :-!.
propcheia( cada(,all(X,S same U,C)), not all_(C,
                                             al_(nil,S same U,C),
                                             E,_,L)) :-!.
propcheia( S, S ) :-!.

aus( nil, cada(X,SS), S, cada(X,(NSS,S)) ) :- sem_true(SS,NSS), !.
aus( SUP+(E-V), SS, S, sup( VS, UV, ( al_(X,(NSS,S),C),F,L ) ) ) :-
    v(E,U,UV),
    sem_cada(SS,SSI), sem_sup(SUP,C,F,VS), sem_sup(SSI,NSS,VS,L), !.
aus( V, cada(X,SS), S, al_(M,NSS,_) ) :- evid(V,UV,C),
    semmesmo(SS,S,C=VV,NB;E,EX,V,M),
    evidencia_(X,EX,E), !.

aus( _, SS, S, (NSS,S) ) :- sem_true(SS,NSS).

sem_true( (A,true), NA ) :- nonvar(A), !, sem_true(A,NA).
sem_true( A, A).

supsn(A+B,FI,F) :- !, aus(A+B,FI,true,F).

```

```

supsn( _ , F,F).

semSUP( ( (ALL,length(X,N)),true) , ALL , V , length(V,N) ).
semSUP( S , S , _ , true ).

sem_SUP(maior,C,max(C,M),M).
sem_SUP(menor,C,min(C,M),M).

semmesmo( ( (al_(_,B,C),length(C,N)),true) , S , VV ,
           (al_(nil,(B,S) same VV,C),length(C,N)) , _ , _ , V , N:V ).

semmesmo( SS , S , VV , al_(EX,(SS,S) same VV,C) , E , EX , V , V:(E=C) ).

sem_cada( cada( _,SC) , SC ) :- !.
sem_cada( _ SC , SC ).

extrmesmo( ( _:_V ):nil,EX, _:cada( SV,SA) ,
           all_(S,al_(nil,(NSV,SA) same V,S),[_]) ) :- var(EX),
                                                    sem_cada(SV,NSV), !.
extrmesmo( VE:VE, _ , _ , SP,SP ).

/* ----- semantica dos quantificadores ----- */

gdef((o:_)-p,cada([XN],F),F,XN). /* possivel all_ ver cada_cada */
gdef((o:_)-s,F,F,_).

ge(nil-nil,nil) :- !.
ge( E1-E2 , E ) :- ge(E1,NE1), ge(E2,NE2), E1-E2\=NE1-NE2, !, ge(NE1-NE2,E).
ge( nil-E , E ) :- !.
ge( E-nil , E ) :- !.
ge( X , X ).

/* ----- controle e semantica de : CADA SN DE CADA SN DE .. ----- */

cada_cada( cada(L,F), M, F ) :- conc(L,M,NM), cada_cada(F,NM,F), !.

/*
cada_cada( all_(L,F) , M, F ) :-
( M=[], cada_cada(F,L,F) ; cada_cada(F,M,F) ), !.
*/

cada_cada( C , [], C ) :- !.
cada_cada( C , M , cada(M,C) ).

/* ----- semantica do complemento do substantivo ----- */

quant_( v(ST, (al_( MN, ( ST ,REL,SV), S) , Length) ) , C,
        (al_( MN, ((C,ST),REL,SV), S) , Length) ) :- !.
quant_( ST , C , (C,ST) ).

/* ----- semantica das relativas ----- */

ordem(cada(Q,(R,S)),SRI,cada(Q,R),(SRI,S)).
ordem((R,S),SRI,R,(SRI,S)).

quant( v( _,true ) , REL, true, cada(Q,SD), cada(Q,SDI)) :-
ultimo(SD,(C,_(all(_(C,_,REL),_)_media)),SRI).

```

```
quant( v( _, ALL ) , REL, SR , SDI, SD ) :-  
quant( ALL, REL, SR, SDI, SD ).
```

```
quant( al_( _, ( _, REL, _ ), _ ), _ ) , REL, true, SD, SDI ).
```

```
quant( _ , SR , SR , SD, SDI ).
```

```
ultimo((A,C,((D,All),true)), (C,D,All), (A,All) ) :- !.
```

```
ultimo((A,B), P, (A,SDI)) :- ultimo(B,P,SDI).
```

```
cuJ(cada(M,F),C,cada(M,(C,F))) :- !.
```

```
cuJ( F , C, (C,F) ).
```

```
cuJo_(cada( _, (A,B) ), not (A,not B)) :- !.
```

```
cuJo_(SR,SR).
```

```
/* ===== MISCELANEA E MENSAGENS ===== */
```

```
in(X,X) :- !.  
in(X,X+L) :- !.  
in(X,Y+L) :- in(X,L).
```

```
member(X,[X|_]).  
member(X,[_|_]) :- member(X,_).
```

```
conc([],L,L).  
conc([X|L1],L2,[X|L3]) :- conc(L1,L2,L3).
```

```
Junta(X) :- X, ! ; assertz(X).
```

```
acordo(X,Y,ser(_),B) :- !, acordo(X,Y,ser:ser,B).  
acordo(X,Y,A,B) :- ( acordo ; X=Y ), ! ;  
EX=(_:core2( _,_,_)), Junta( mens(saída(facc(A,B))) ), fail.
```

```
facc(A,B) :- saída(mess(s13)),  
print_(A), write(' e '), print_(B), put(46).
```

```
comp(ENOM,ECOMP) :- write(' '), write(ECOMP), put  
saída(mess(s11)), write(ENOM), put saída(mess(s12)).
```

```
resf(M) :- write('> '), print_(M), fail(2).  
resf(M) :- print_(M), nl, nl.
```

```
print_(acento(X)) :- write(X), write(''), !.  
print_(_:X_) :- write(X), !.  
print_(_:X) :- write(X), !.  
print_(X_) :- write(X), !.  
print_( X ) :- write(X).
```

```
/* =====  
/* ===== */
```

SINTAXE E SEMÂNTICA ESPECIFICAS


```
/* **** [ GRA.FL ] **** */
```

```
- op(30,xfy, - ).  
- op(35,xfx, : ).
```

```
/* ===== GRAMATICA ORBI ===== */
```

```
/* --- frase : (o) que + e + snb --- */
```

```
/*  
frase(LE,que_e(XN),LS) :-  
  c_que_e(LE,L1),  
  snb(L1,pos:_,_,XN,_,_,nil,LS).
```

```
c_que_e([o;o,_,_],que+P(que),EIL3,L) :- /* /* c que + e */  
/* in(v(ser:_,_,_),E), !.
```

```
c_que_e([que+P(que),EIL3,L) :- /* /* que + e */  
/* in(v(ser:_,_,_),E). */
```

```
/* --- frase : porque + sn + prop --- */
```

```
/*  
frase(LE,porque(NSF),LS) :-  
  extraf(LE,[porque|L1],EX,EXS1),  
  extraf(L1,L2,EX,EXS2),  
  sn(L2,FD,AS,MS,SS,STS,E,EX,SK,L3),  
  props(suj(AS):dn(FD,MS,SS,STS,_),E:(S,SV),EX,L3,LS),  
  sus(E,S,SV,SP),  
  propeia(SF,NSF).
```

```
*/
```

```
comen(NT,ST,X,AF,NSP,VES-VE,EX,LCOMP,LE,LS) :- comlcomp(NT),  
  pccao(NT,LE,LC),  
  contrac(F(F),LO,L1),  
  member(F,[de,para,em]),  
  snb(L1,FD,AC,NC,_,SI,F,VE,L2),  
  postcom(NT,NC,ST,SC,C),  
  quant_(ST,C,NST),  
  write3(NC,SC,X,NST,AF,F,VE,EX,s(LCOMP,L2,NT,SI,FD,AC,NSP,LS,LC)),  
  write3(NC,SC,X,NST,AF,F,VE,EX,s(LCOMP|LCOMP,LC,NT,SI,FD,AC,NSP,LS,LC)),  
  comen(NC,SC,X,(NST,AF),F,VE,EX,LCOMP,L2,L3),  
  C(L3=LC, L2=L3, C=comen(NT,ST,X,AF,NSP,VES-VE,EX,LCOMP,LE,LS),  
    NT=media:L, L2=L3, S=sn ),  
  cadadecada(SB,[],S1),  
  relats(L3,suj(AC):dn(FD,NC,SI,SC,LC),NSP,EX,O,LS) :-  
    NSP=SI,  
    L3=LS  
  ),  
  conc(COMP,L1,LC). /*para ellipse*/
```

```
pccao(media:L,Cconc(F(de)+d(I,S-P,def-Pos)),nc(valor:T-L,AN)|L0,LC) :- !,  
  acordo(S-P,AN,I,valor:T).
```

```
pccao(L,L,L).
```

```
comlcomp(_). /* se nao ha informacao em contrario */
```

```
/* ---- sn:sujeito + sv + pres + <sn:complemento> ---- */
```

```

PROP(comp(TC):dn(L,MC,SC,STC,L),VE:(NSS,NSC,NSV),EX,LE,LS) :-
  sn(LE,FD,AS,MS,SS,STS,L,EX,L1),
  sv(L1,FD,AV,VE,MV,L,TC,FV,EX,LS), MV\=(ser(L):L),
  testars(MV,MS,MC,(STS,STC),SV),
  acordo(AS,AV,MS,MV),
  auscomp(SS,NSS,MS,MC),
  sem_cada(SC,NSC),
  forma_v(FV,SV,NSV).

```

/* --- <sn_sujeito> + sv + prep + sn_complemento --- */

```

PROP(suj(AS):dn(FD,MS,SS,STS,L),VE:(SS,SCV),EX,LE,LS) :-
  sv(LE,FD,AV,L,MV,L,TC,FV,EX,L1),
  prep(TC,L1,L2),
  sn(L2,L,L,MC,SC,STC,VESN,EX,L,LS),
  testars(MV,MS,MC,(STS,STC),SV),
  solve((MV=(UB:MA), VESN=nil, sup(MA,MS,VE) ; VE=VESN)),
  acordo(AS,AV,MS,MV),
  forma_v(FV,(SC,SV),SCV).

```

TC = r(que).^v

/* --- <sn_sujeito> + ser/adJ /existir/haver --- */

```

PROP(suj(AS):dn(FD,MS,SS,STS,L),VE:(SS,NSVA),EX,LE,LS) :-
  sv(LE,FD,AA,VE,VB:MA,adJ,obj,FV,EX,LS);
  testars(VB:MA,MS,L,SVAI),
  solve((sup(MA,MS,VE), SVA=true ; VE=nil, SVA=SVAI)),
  acordo(AS,AA,MS,MA),
  forma_v(FV,(true,SVA),NSVA).

```

/* --- <sn_complemento> + sv + prep + sn_sujeito --- */

```

PROP(comp(TC):dn(FD,MC,SC,STC,L),nil:(NSS,NSC,NSV),EX,LE,LS) :-
  sv(LE,FD,AV,L,MV,L,TC,FV,EX,L1), MV\=(ser(L):L),
  TC\F(F), prep(F(F),L1,L2);
  L1=L2),
  sn(L2,FD,AS,MS,SS,STS,L,EX,L,LS),
  testars(MV,MS,MC,(STS,STC),SV),
  acordo(AS,AV,MS,MV),
  auscomp(SS,NSS,MS,MC),
  sem_cada(SC,NSC),
  forma_v(FV,SV,NSV).

```

/* --- SN = entre SN1 e SN2 --- */

```

sn([F(entre) | LE],L,L,(MN1,MN2),L,(ST1,ST2),L,L,LS) :-
  sn(LE,L,L,MN1,L,ST1,L,L,L1),
  conj(e,L1,L2),
  sn(L2,L,L,MN2,L,ST2,L,L,L2).

```

/* --- SN = 0/1/2/3/.. OU nao/muito/pouco apto em ; ser/valer/superior/inferior a/.. + SN --- */

```

sn(LE,L,L,valor:valor-(SN-Z),(v(X,igual(Y,SN)),true),v(X,igual(Y,SN)),
nil,L,ESN+[0][0],LS) :-
  sn_valor(LE,LS,SN),
  (integer(SN), ESN=n(SN) ;
  ESN=SN), !.

```

/* --- sintagma de base (S:F) --- */

```

snb([DIL1],Pos:o,GN-NN,XN,_,SD,F,nil,LS) :- /* (det) + nc + goft x
(( D=det, MD=o:_, ND=NN /* DU (det) + nca + no x
in(d(MDI,GI-ND,_-Pos),D) ),
(L1=[nc(XN,GN-NN)IL2], MD=MDI, /* o(s)/a(s) + nc + goft x
goft(L2,XN,LS) /* DU o(s)/a(s) + nca + no x
L1=[NCAIL2],
in(nca(XN,GN-NN),NCA), MD-ND=(o:_) -s,
na([XNIL2],LS) );
in(nca(XN,GN-NN),D), /* DU nca + na */
na([XNIL1],LS), MD-ND=(o:_) -s ),
acordo(GI-ND,GN-NN,MD,XN),
gdef(MD-ND,SD,F,XN).

```

/* --- designacao dum substantivo --- */

```

goft([XILE],XN,LS) :-
in(nca(MN,_) ,X), /* nca + na ex: D 6b x
na([XNILE],LS),
(MN=XN, ! ;
Junta(mens(saída((mess(s5), print_(MN), mesf('.'))))),
fail ).

```

```

goft(L,XN,LS) :- /* na ex: 6b x
na([XNIL],LS): !.

```

```

goft(LS,_,LS).

```

/* --- os substantivos e designacoes --- */

```

na([ponto: _-X/Y,n(X),virs,n(Y)IL],L).

```

```

na([resiao: _-(W,X,Y,Z;n(W),virs,n(X),pontoevirs,n(Y),virs,n(Z)IL],L).

```

```

na([C,n(A)IL],L) :- in(artidao:_-(A-_),C).

```

```

na([factor: _-(F-_),n(N)ILE],LS) :- design(LE,N,F,LS).

```

```

na([descriptor: _-(D-_),n(N)ILE],LS) :- design(LE,N,D,LS).

```

```

na([valor: _-(V-_)ILE],LS) :-

```

```

sn_valor(LE,LS,V) ;

```

```

LE=LS.

```

```

na([_: _-(F-_),meta([F])IL],L).

```

```

sn_valor([EIL],L,V) :- member(E,[n(V),b,c,r,s,h,i,j,k]),

```

```

try((E=n(V),ex(V))), !.

```

```

sn_valor([F,artoIL],L,[F,arto]) :- member(F,[nac,poqco,quito]), !.

```

```

sn_valor([LTIL],L,V) :-

```

```

member(Lt:V,[arto:[arto],

```

```

(_+(p:)+_):a,

```

```

nca(factor:_,_+f,

```

```

nca(descriptor:_,_):o]), !.

```

```

design(LE,N,a(N,NQ),LS) :- sn_valor(LE,LS,V), atom(V), name(V,[NQ]), !.

```

```

design(L,N,N,L).

```

/* --- relativa CUJ(DIA) cu ONDE --- */

```

relat(LE, _dn(AS, MS, SS, SD, NF, E, L2), E-EP:(S, SR), EX, LS) :-
  opcao_cuj(LE, L1),
  snb(L1, pos:AS, MS, SS, SD, NF, E, L2),
  compn(MS, SS, NX-SX, true, F, EX, L2, L3),
  testcomp(NX, MNR, SX, SNR, C),
  cuJ(F, C, NF),
  cadecada(SD, [], NSD),
  relat_cont(EP, AS, MS, NSD, SS, EX, L3, LS, SR),
  relat_cont(EP, AS, MS, NSD, SS, EX, L3, LS, SR) :-
  props(suj(AS):dn(pos:AS, MS, NSD, SS, IJ), EP:(NSD, SV), EX, L3, LS),
  aus(nil, NSD, SV, SRI),
  cuJo_(SRI, SR).

```

```

opcao_cuj([conc(cujo+D)ILO], [D|ILO]).
opcao_cuj([onde|L1], L1).
opcao_cuj([P(em), que+_IL1], L1).

```

/* --- relativa : ... existe(m) + rel -> rel + ha --- */

```

relats([v(existir:T, AV, TC)IL], suj(AN):dn(FD, MN, SD, ST, []), NSD, EX, 0,
  [v(haver:T, AV, TC)IL1]) :-
  acordo(AN, AV, existir:T, MN),
  relats(L, suj(AN):dn(FD, MN, SD, ST, []), NSD, EX, 0, L1),
  L1\L.

```

/* --- relativa : que + existe(m) --- */

```

relat([que+_v(existir:T, AV, _)IL], suj(AN):dn(_, MN, SD, _, []),
  nil:(SD, true), _, L) :-
  acordo(AN, AV, MN, existir:T),

```

```

relats([v(existir:T, AV, TC)IL],
  suj(AS):dn(FD:_, MS, SS, SBS, RS), NSD, EX, 0, [v(haver:T, AV, TC)IL1]) :-
  acordo(AS, AV, existir:T, MN),
  relats(L, suj(AS):dn(FD:rel, MS, SS, SBS, RS), NSD, EX, 0, L1),
  L1\L.

```

/****** SEMANTICA *****/

/* --- semantica igual --- */

```

sem_v(F, P) :- nonvar(F), F=v(_, igual(_, _)), !.

```

/* --- Semantica nome sem complementos --- */

```

nome(NT, ST) :- testcomp(NT, _, ST, _, _).

```

~~/* --- Semantica OFALL : testcomp & testarg(ter) --- */~~

```

testcomp(media: _-M-, COMP,
  (all(V:(D, SCOMP, SV):S), media(S, M)), v(SCOMP, true), true):-!,
  testcomp(valor:_, COMP, v(_igual(V, _)), SCOMP, true).
testcomp(TNDM:ENDM-ANDM, TCOMP:ECOMP-ACOMP, S, SC, C) :-
  (forall(TNDM:ANDM, TCOMP:ACOMP, S, SC, C)
  ;
  sem_v(S, ENDM),

```

```
sem_v(SC,SCOMP),
ofall(TNOM:ANOM,TCOMP:ACOMP,SNOM,SCOMP, C) ), ! ;
```

```
Juntar(mens(saida(fcpn(ENOM,ECOMP))), fail.
```

```
ofall( ponto:(X/Y) , resiao:(M, -D,F),
      f(X,Y,D) , f(M,N,D,F) , true).
```

```
ofall( descriptor:(DE-V) , M ,
      d(DE,D,V) , S , C ) :- tr(M,D),
      member(M+S+C,[ponto:(X/Y)+f(X,Y,D)+true,
                    (factor:(X-Y))+f(X,D,Y)+em(d(DE,D,_),f(X,D,_)),
                    (aptidao:(X-Y))+a(X,D,Y)+( em(f(A,D,_), a(X,D,_)),
                    em(d(DE,D,_),f(A,D,_)))]).
```

```
ofall( factor:(F-V) , M ,
      f(F,D,V) , S , C ) :- tr(M,D),
      member(M+S+C,[ponto:(X/Y)+f(X,Y,D)+true,
                    (aptidao:(X-Y))+a(X,D,Y)+em(f(F,D,_),a(X,D,_))]).
```

```
ofall( aptidao:(A-V) , M ,
      a(A,D,V) , f(X,Y,D) , true ) :- tr(M,D).
```

```
ofall( representatividade:R , M ,
      v(D,igual(R,X)) , S, true ) :- V=(L-R),
      member(M+S,[aptidao:(A-V)+a(A,D,V),
                    (factor:(A-V))+f(A,D,V),
                    (descriptor:(A-V))+d(A,D,V)]).
```

```
ofall( valor:(X-R) , M ,
      v(D,igual(N,X)) , S , true ) :- V=(N-R),
      member(M+S,[aptidao:(A-V)+a(A,D,V),
                    (factor:(A-V))+f(A,D,V),
                    (descriptor:(A-V))+d(A,D,V),
                    (representatividade:R)+_,
                    (media:V)+_,
                    (valor:V)+_]).
```

```
ofall( resiao:(M,N,D,F) , _ ,
      r(M,N,D,F) , _ , true ).
```

```
ofall( resta:(L--A1), M:(N--nil) ,
      (T, des(N,NN),
      A1=representar((H:-B)), /* para as elices */
      clause(H,F) , _ , true ) :-
      ( var(M), T=member(H,[a(N,X,Y),f(N,X,Y)]) ;
      member(M+H,[aptidao+a(NN,X,Y),factor+f(NN,X,Y)]), T=true).
```

```
tr(X:_,d(_,_,_)) :- X==ponto.
tr(_,_).
```

```
/* --- semantica ser --- */
```

```
ser( igual , X , V, igual(V,X) ).
ser( entre , (X,Y) , V, (compa(V)=X),compa(V)=Y) ).
ser( diferente , X , V, V\=X ).
ser( menor , X , V, compa(V<X) ).
ser( maior , X , V, compa(V>X) ).
```

```
/* --- teste valor --- */
```

```
fval(N) :- val(K), ! ; Juntar(mens(saida(( mens(N),mes(sé) )) ), fail.
```

```

val(representatividade).
val(media).
val(valor).
val(aptidao).
val(factor).
val(descriptor).

/* --- teste valor < ou = a 5 --- */
ex(X) :- X<=5, ! ; Juntar(mens(saida(nao_ex(valor,X)))), fail.
nao_ex(D,V) :- saida((mess(s7), write(D), tab(1), write(V))).

/* --- acesso a variavel com valor --- */
v( representatividade, V, V ).
v( valor, V_ , V ).
v( _ , _-V_ , V ).
v( _ , _ , _ ).

/* ----- controle dos argumentos do verbo e semantica da relacao -----
testars( ter:_ , ponto:_ , valor:_ , (P(_,_),v(D,_)) , true ) :- var(D), !.
testars( ter:_ , TMS:EMS-AMS , TMC:EMC-AMC , (SS,SC) , F ) :-
    sem_v(SS,NSS),
    sem_v(SC,NSC),
    ofall(TMC:AMC,TMS:AMS,NSC,NSS,F), ! ;
    Juntar(mens(saida(factor(EMC,EMS))), fail.
testars( ser(_):entre , MS:_-VS , (MC1:_-VC1,MC2:_-VC2), _ , F ) :-
    v(MS,VS,V1), v(MC1,VC1,V2), v(MC2,VC2,V3),
    fval(MS), fval(MC1), fval(MC2),
    ser(entre,(V2,V3),V1,F), !.
testars( ser(VA):ADJ , MS:_-VS , MC:_-VC , (SS,SC) , (isual(V1,VA),F) ) :-
    v(MS,VS,V1), v(MC,VC,V2),
    fval(MS), ( MC=ponto ; fval(MC) ),
    ser(ADJ,V2,V1,F),
    ( true((nonvar(VA),ofall(MS:VS,MC:VC,SS,SC,_))) ;
      true((var(VA),nonvar(SS),nonvar(SC),
        ars(2,SS,A),ars(2,SC,A))) ) ;
testars( V:T , _ , h , _ , true) :- ( V=existir ; V=haver ),true(M=existir:T

auscomp( cada(A,E), cada(A,E) , _ , _ ).
auscomp( SS , cada([MS,MC],SS) , MS, MC).

/* =====
=====

```

CONHECIMENTO METALINGUISTICO

/* ##### [Palavras] ##### */

/* ===== BASE DE DADOS METALINGUISTICOS : AS PALAVRAS =====

/* ----- PARTE NUCLEAR ----- */

Palavra(' ---- NUCLEAR ----'),

Palavra(o-os),

Palavra(uma-umas),

Palavra(cada),

Palavra(todo-todos),

Palavra(toda-todas),

Palavra(qualquer-qualsquer),

Palavra('0,1,2,3,..'),

Palavra(nenhum-nenhuns),

Palavra(nenhuma-nenhumas),

Palavra(algum-alsuns),

Palavra(alguma-alsumas).

Palavra(de).

Palavra(em).

Palavra(com).

Palavra(sem).

Palavra(para).

Palavra('entre .. e ..').

Palavra(ao-aos).

Palavra(as).

Palavra(na-nas).

Palavra(no-nos).

Palavra(numa-numas).

Palavra(num-nuns).

Palavra(da-das).

Palavra(do-dos).

Palavra(da-da-dumas).

Palavra(da-da-duns).

Palavra(e).

Palavra(ou).

Palavra(que).

Palavra(cujo).

Palavra(cuja).

Palavra(porque).

Palavra(qual-queis).

Palavra(quanto-quantos).

Palavra(quanta-quantas).

Palavra('e que').

Palavra('sera que').

Palavra('de-me').

Palavra('de-me').

Palavra(nao).

Palavra('e''-sao).

Palavra(tem-'te'm').

Palavra(vale-valem).

Palavra(existe-existem).

Palavra('ha'').

/* ----- PARTE ORBI ----- */

Palavra(' '),
Palavra(' ---- ORBI ---- '),
Palavra(resiao-resioes),
Palavra(ponto-pontos),
Palavra(aptidao-aptidoes),
Palavra(factor-factores),
Palavra(descriptor-descriptores),
Palavra(representatividade-representatividades),
Palavra(valor-valores),
Palavra(media-medias),

Palavra(apto),
Palavra(mesmo-mesmos),
Palavra(mesma-mesmas),
Palavra('igual a'-iguais),
Palavra('diferente de'-diferentes),
Palavra('inferior a'-inferiores),
Palavra('superior a'-superiores),
Palavra('maior que'-maiores),
Palavra('menor que'-menores),

Palavra('A'-aptidao),
Palavra('F'-factor),
Palavra('D'-descriptor),
Palavra('P'-ponto),
Palavra('RG'-resiao),
Palavra('RP'-representatividade),

Palavra(pouco),
Palavra(muito).

/* ----- PARTE METALINGUISTICA ----- */

Palavra(' '),
Palavra(' ---- METALINGUAGEM ---- '),
Palavra(vocabulario-vocabularios),
Palavra(palavra-palavras),
Palavra(nome-nomes),
Palavra(verbo-verbos),
Palavra(artigo-artigos),
Palavra(adjectivo-adjectivos),
Palavra(preposicao-preposicoes),
Palavra(pronome-pronomes),
Palavra(contraccao-contraccoes),
Palavra(conjuncao-conjuncoes),
Palavra(adverbio-adverbios),
Palavra(locucao-locucoes),
Palavra(sujeito-sujeitos),
Palavra(complemento-complementos),
Palavra(abreviatura-abreviaturas).

/* =====
===== */

/* ----- semantica dos nomes sem complemento ----- */

nome(palavra: _-F, palavra_(F)).
nome(vocabulario: _-F, palavra_(F)).
nome(nome: _-N, nome_(N,_,_)).
nome(artigo: _-A, artigo_(A)).
nome(adjectivo: _-A, adjectivo_(A)).
nome(preposicao: _-P, preposicao_(F)).
nome(contraccao: _-C, contraccao_(C)).
nome(conjuncao: _-C, conjuncao_(C)).
nome(pronome: _-Pronome-F, pronome_(F)).
nome(pronome: _-relativ-F, pronome_relativo(F)).
nome(pronome: _-interro-F, pronome_interrosativo(F)).
nome(locucao: _-L, locucao_(L)).
nome(adverbio: _-A, adverbio_(A)).
nome(abreviatura: _-A, abreviatura_(A)).
/* nome(verbo: _-V, verbo(V)). */

/* =====
===== BASE DE DADOS METALINGUISTICOS =====
=====

/* ----- PARTE NUCLEAR ----- */

palavra_(A) :- artigo_(A).
palavra_(F) :- preposicao_(F).
palavra_(C) :- contraccao_(C).
palavra_(C) :- conjuncao_(C).
palavra_(F) :- pronome_(F).
palavra_(L) :- locucao_(L).
palavra_(A) :- adverbio_(A).
palavra_(A) :- adjectivo_(A).
palavra_(N) :- nome_(N,_,_).
palavra_(A) :- abreviatura_(A).
/* palavra_(V) :- verbo_(V). */

artigo_(a-as).
artigo_(o-os).
artigo_(uma-umas).
artigo_(cada-)).
artigo_(todo-todos).
artigo_(toda-todas).
artigo_(qualquer-qualsquer).
artigo_(1-'2,3,...').
artigo_(nenhum-nenhuns).
artigo_(nenhuma-nenhumas).
artigo_(algum-alsuns).
artigo_(alguma-alsumas).

preposicao_(de).
preposicao_(em).
preposicao_(com).
preposicao_(sem).
preposicao_(para).
preposicao_('entre .. e ..').

contraccao_(achados).
contraccao_(as-).
contraccao_(ca-)

contraccão_(no-nos).
contraccão_(numa-numas).
contraccão_(num-nuns).
contraccão_(da-das).
contraccão_(do-dos).
contraccão_(duma-dumas).
contraccão_(dum-duns).

conjunção_(e).
conjunção_(ou).

pronome_(F) :- pronome_relativo(F).
pronome_(F) :- pronome_interrogativo(F).

pronome_relativo(que).
pronome_relativo(cujo).
pronome_relativo(cuja).

pronome_interrogativo(que-que).
pronome_interrogativo(porque-).
pronome_interrogativo(qual-quais).
pronome_interrogativo(quanto-quantos).
pronome_interrogativo(quanta-quantas).

locução_('e que').
locução_('sera que').
locução_('da-me').
locução_('de-me').

advérbio_(não).

nome_(vocabulario-vocabularios, [], []).
nome_(palavra-palavras, [], []).
nome_(nome-nomes, [], []).
nome_(verbo-verbos, [], []).
nome_(artiso-artisos, [], []).
nome_(adjectivo-adjectivos, [], []).
nome_(preposicao-preposicoes, [], []).
nome_(pronome-pronomes, [], []).
nome_(contraccão-contraccoes, [], []).
nome_(conjunção-conjuncoes, [], []).
nome_(advérbio-advérbios, [], []).
nome_(locução-locucoes, [], []).
nome_(sujeito-sujeitos, [], [verbo-verbos]).
nome_(complemento-complementos, [], [nome-nomes, verbo-verbos]).
nome_(abreviatura-abreviaturas, [], [artido-artidos, factor-factores,
descriptor-descriptores, representatividade-representatividades,
ponto-pontos, resiao-regioes]).

adjectivo_(relativo-relativos).
adjectivo_(interrogativo-interrogativos).

/* ----- PARTE ORBI ----- */

nome_(resiao-regioes, ['W,X;Y,Z'], []).

nome_(ponto-pontos, ['X,Y'],
[resiao-regioes]).

nome_(artido-artidos, [1,2,3,4,5],

nome_(factor-factores, [1,2,3,4,5,6,7,8,9,10,11],
[aptidao-aptidoes,ponto-pontos]).

nome_(descriptor-descriptores, [1,2,3,4,5,6,7,8,9,10],
[factor-factores,aptidao-aptidoes,ponto-pontos,regiao-regioes]).

nome_(representatividade-representatividades, [1,2,3,4,5],
[descriptor-descriptores,factor-factores,aptidao-aptidoes]).

nome_(valor-valores, [1,2,3,4,5],
[descriptor-descriptores,factor-factores,aptidao-aptidoes,
representatividade-representatividades,media-medias]).

nome_(media-medias, [],
[valor-valores,descriptor-descriptores,factor-factores,
aptidao-aptidoes,representatividade-representatividades]).

adjectivo_(apto-).

adjectivo_(mesmo-mesmos).

adjectivo_(mesma-mesmas).

adjectivo_('igual a'-iguais).

adjectivo_('diferente de'-diferentes).

adjectivo_('inferior a'-inferiores).

adjectivo_('superior a'-superiores).

adjectivo_('maior que'-maiores).

adjectivo_('menor que'-menores).

abreviatura_('A'-aptidao).

abreviatura_('F'-factor).

abreviatura_('D'-descriptor).

abreviatura_('P'-ponto).

abreviatura_('RG'-regiao).

abreviatura_('RP'-representatividade).

adverbio_(pouco).

adverbio_(muito).

```

/* #### ANALYS.PL #### */

?- op(30,xfy,-).

/* ----- entrees/sorties ----- */

f(P) :- n(1,P,NP),
        ph(NP),
        write('> Et maintenant efface ! (a,b,c)'),
        nl.

n(N,[],[]) :- !.

n(N,[MILM],[M-NINLM]) :-
    NN is N+1,
    n(NN,LM,NLM).

ph(P) :-
    assert(mot(0)),
    assert(analyse(1)),
    assert(analyse(1,0)),
    ( ph(P,[]),
      write('> Acceptee !'), nl ;
      write('> Rejetee avec '),
      retract(analyse(N)),
      write(N), write(' suite(s) analysable(s)'), nl,
      parties(N,P)
    ),
    nl, nl.

a :- retract(mot(X)), write('> dernier mot : '), write(X), nl, fail.
b :- retract(analyse(X)), write('> derniere analyse : '), write(X), nl, fail.
c :- retract(analyse(X,Y)), write('          analyse '), write(X),
     write(', avec '), write(Y), write(' comme dernier mot'), nl, fail.

parties(0,_) :- !.

parties(N,P) :-
    write(' - suite '), write(' : '),
    retract(analyse(N,NM)),
    partie(NM,P),
    NN is N-1,
    parties(NN,P).

partie(N,[M-NI_L]) :- !, write(' '), write(M), nl.

partie(N,[M-LIL]) :- write(' '), write(M), partie(N,L).

/* ----- trace de l'analyse ----- */

t(N) :-
    retract(mot(NNM)),
    assert(mot(N)),
    ( N>NNM, !,
      analyse(NNA),
      retract(analyse(NNA,NNM)),
      assert(analyse(NNA,N);
      N<NNM, !,
      retract(analyse(NNA)),
      NNNA is NNA+1,
      assert(analyse(NNNA)),
      assert(analyse(NNNA,NNM))
    ).

```

/* ----- grammaire ----- */

ph(LE,LS) :-
 sn(LE,L1),
 sv(L1,LS).

ph(LE,LS) :-
 sn(LE,LS).

sn(LE,LS) :- snb(LE,L1), rel(L1,LS).

rel([qui-NILE],LS) :-
 t(N),
 sv(LE,LS).

rel(L,L).

sv([aime-NILE],LS) :- t(N), sn(LE,LS).
sv([connait-NILE],LS) :- t(N), sn(LE,LS).
sv([croit-N,que-N1ILE],LS) :- t(N), t(N1), ph(LE,LS).
sv([croit-NILE],LS) :- t(N), sn(LE,LS).
sv([pense-N,a-N1ILE],LS) :- t(N), t(N1), sn(LE,LS).
sv([pense-N,que-N1ILE],LS) :- t(N), t(N1), ph(LE,LS).
sv([pense-NILE],LS) :- t(N), LE=LS.

snb([Jean-NILE],LS) :- t(N), LE=LS.
snb([Isor-NILE],LS) :- t(N), LE=LS.
snb([Mary-NILE],LS) :- t(N), LE=LS.
snb([Jane-NILE],LS) :- t(N), LE=LS.

/* ##### [dilins] ##### */

?-op(30,xfy,-).
?-op(35,xfx,!).

/* ===== DICCIONARIO METALINGUISTICO =====

vocabulario(nc(vocabulario:vocabulario-_,m-s)),
vocabularios(nc(vocabulario:vocabularios-_,m-p)),

palavra(nc(palavra:palavra-_,f-s)),
palavras(nc(palavra:palavras-_,f-p)),

nome(nc(nome:nome-_,m-s)),
nomes(nc(nome:nomes-_,m-p)),

verbo(nc(verbo:verbo-_,m-s)),
verbos(nc(verbo:verbos-_,m-p)),

artigo(nc(artigo:artigo-_,m-s)),
artigos(nc(artigo:artigos-_,m-p)),

adjetivo(nc(adjetivo:adjetivo-_,m-s)),
adjetivos(nc(adjetivo:adjetivos-_,m-p)),

preposicao(nc(preposicao:preposicao-_,f-s)),
preposicoes(nc(preposicao:preposicoes-_,f-p)),

pronome(nc(pronome:pronome-_,m-s)),
pronomes(nc(pronome:pronomes-_,m-p)),

contracao(nc(contracao:contracao-_,f-s)),
contraccoes(nc(contracao:contraccoes-_,f-p)),

conjuncao(nc(conjuncao:conjuncao-_,f-s)),
conjuncoes(nc(conjuncao:conjuncoes-_,f-p)),

locucao(nc(locucao:locucao-_,f-s)),
locucoes(nc(locucao:locucoes-_,f-p)),

abreviatura(nc(abreviatura:abreviatura-_,f-s)),
abreviaturas(nc(abreviatura:abreviaturas-_,f-p)),

sujeito(nc(sujeito:sujeito-_,m-s)),
sujeitos(nc(sujeito:sujeitos-_,m-p)),

complemento(nc(complemento:complemento-_,m-s)),
complementos(nc(complemento:complementos-_,m-p)),

adverbio(nc(adverbio:adverbio-_,m-s)),
adverbios(nc(adverbio:adverbios-_,m-p)),

/* =====
=====

```
/* ##### [ selins ] ##### */
```

```
?-op(30,xfy,-).  
?-op(35,xfx,!).
```

```
/* ===== SEMANTICA METALINGUISTICA =====
```

```
/* ----- semantica dos nomes sem complemento ----- */
```

```
ofall( palavra:F      , - , lins(palavra_(F))      , - , true ).  
ofall( vocabulario:F  , - , lins(palavra_(F))      , - , true ).  
ofall( artigo:A      , - , lins(artigo_(A))      , - , true ).  
ofall( nome:N        , - , lins(nome_(N,_))      , - , true ).  
ofall( adjetivo:A    , - , lins(adjectivo_(A))   , - , true ).  
ofall( preposicao:F   , - , lins(preposicao_(F))   , - , true ).  
ofall( contraccao:C  , - , lins(contraccao_(C))  , - , true ).  
ofall( conjuncao:C   , - , lins(conjuncao_(C))   , - , true ).  
ofall( pronome:F     , - , lins(pronome_(F))     , - , true ).  
ofall( locucao:L     , - , lins(locucao_(L))     , - , true ).  
ofall( adverbio:A    , - , lins(adverbio_(A))    , - , true ).  
ofall( abreviatura:A , - , lins(abreviatura_(A)) , - , true ).  
ofall( verbo:V      , - , lins(verbo_(V,_,_))   , - , true ).
```

```
ofall( complemento:C, nome:N      ,  
      true          , lins(nome_(N,C)), true ).
```

```
ofall( complemento:C, verbo:V    ,  
      true          , lins(verbo_(V,_C)), true ).
```

```
ofall( sujeito:S, verbo:V      ,  
      true       , lins(verbo_(V,S,_)), true ).
```

```
/* =====  
=====
```


/* ##### [balins] ##### */

/* ===== BASE DE DADOS METALINGUISTICOS =====

artiso_(o-os).
artiso_(uma-umas).
artiso_(cada).
artiso_(todo-todos).
artiso_(toda-todas).
artiso_(qualquer-qualsquer).
artiso_('0,1,2,3,..').
artiso_(nenhum-nenhuns).
artiso_(nenhuma-nenhumas).
artiso_(algum-alguns).
artiso_(alguma-algumas).

preposicao_(de).
preposicao_(em).
preposicao_(com).
preposicao_(sem).
preposicao_(para).
preposicao_('entre .. e ..').

contracao_(ao-aos).
contracao_(as).
contracao_(na-nas).
contracao_(no-nos).
contracao_(numa-numas).
contracao_(num-nuns).
contracao_(da-das).
contracao_(do-dos).
contracao_(dum-dumas).
contracao_(dum-duns).

conjuncao_(e).
conjuncao_(cu).

pronome_(que).
pronome_(cujo-cujos).
pronome_(cuja-cujas).
pronome_(qual-quals).
pronome_(quanto-quantos).
pronome_(quanta-quantas).
pronome_(porque).

locucao_('e' que').
locucao_('sera que').
locucao_('da-me').
locucao_('de-me').
locucao_('o que').

nome_(' ---- METALINGUAGEM ---- ', [' ']).
nome_(vocabulario-vocabularios, ['sem complemento']).
nome_(palavra-palavras, ['sem complemento']).
nome_(nome-nomes, ['sem complemento']).
nome_(verbo-verbos, ['sem complemento']).
nome_(artiso-artisos, ['sem complemento']).
nome_(adjetivo-adjetivos, ['sem complemento']).
nome_(preposicao-preposicoes, ['sem complemento']).
nome_(pronome-pronomes, ['sem complemento']).
nome_(contracao-contracoes, ['sem complemento']).
nome_(conjuncao-conjuncoes, ['sem complemento']).

nome_(adverbio-adverbios, ['sem complemento']),
nome_(locucao-locucoes, ['sem complemento']),
nome_(abreviatura-abreviaturas, ['sem complemento']),
nome_(sujeito-sujeitos, [verbo-verbos]),
nome_(complemento-complementos, [nome-nomes, verbo-verbos]).

nome_(' ', [' ']),
nome_(' ---- ORRI ---- ', [' ']),
nome_(regioao-regioes, ['sem complemento']),
nome_(ponto-pontos, [regioao-regioes]),
nome_(aptidao-aptidoes, [ponto-pontos, regioao-regioes]),
nome_(factor-factores, [aptidao-aptidoes, ponto-pontos, regioao-regioes]),
nome_(descriptor-descriptores, [factor-factores, aptidao-aptidoes,
ponto-pontos, regioao-regioes]),
nome_(representatividade-representatividades, [descriptor-descriptores,
factor-factores,
aptidao-aptidoes]),
nome_(valor-valores, [representatividade-representatividades,
descriptor-descriptores, factor-factores,
aptidao-aptidoes, media-medias]),
nome_(media-medias, [valor-valores, representatividade-representatividades,
descriptor-descriptores, factor-factores,
aptidao-aptidoes, media-medias]).

adjectivo_(apto),
adjectivo_(mesmo-mesmos),
adjectivo_(mesma-mesmas),
adjectivo_('igual a'-iguais),
adjectivo_('diferente de'-diferentes),
adjectivo_('inferior a'-inferiores),
adjectivo_('superior a'-superiores),
adjectivo_('maior que'-maiores),
adjectivo_('menor que'-menores).

abreviatura_('A'-aptidao),
abreviatura_('F'-factor),
abreviatura_('D'-descriptor),
abreviatura_('P'-ponto),
abreviatura_('RG'-regiao),
abreviatura_('RP'-representatividade).

adverbio_(nao),
adverbio_(pouco),
adverbio_(muito).

verbo_('e''-sso, ['qualquer nome',
[igual-iguais, superior-superiores, inferior-inferiores, maior-maiores,
menor-menores, 'entre .. e ..', diferente-diferentes, '1,2,...',
apto, 'nao apto', 'pouco apto', 'muito apto', '...']).

verbo_(tem-'te'm', [valor-valores, descriptor-descriptores, factor-factores,
aptidao-aptidoes, representatividade-representatividades,
media-medias, ponto-pontos],
[valor-valores, descriptor-descriptores, factor-factores,
aptidao-aptidoes, representatividade-representatividades,
media-medias]).

verbo_(vale-valem, [valor-valores, descriptor-descriptores,
factor-factores, aptidao-aptidoes,
representatividade-representatividades, media-medias],
['1,2, .., apto, ...']).

verbo_(existe-existem, ['qualquer nome', ['sem complemento']).

```
/* ##### E filings I ##### */
```

```
top(30,xf9,-).  
top(30,12,!).
```

```
/* ===== SEE/SEEN FILES =====
```

```
/* ---- files palins e balins (metalinguagem) ---- */
```

```
ling(palavra_(X)) :- !, see(palins), seen, see(palins),  
                    repeat, read(P),  
                    ( P=(?-end), seen, !, fail ;  
                      P=palavra(X) ).
```

```
ling(X) :- see(balins), seen, see(balins),  
           repeat, read(P),  
           ( P=(?-end), seen, !, fail ;  
             P=X ).
```

```
/* ---- file exalic (explicacoes) ---- */
```

```
e_(X,R) :- !, see(exalic), seen, see(exalic),  
          repeat, read(P),  
          ( P=(?-end), seen, !, fail ;  
            P=(e_(X,R)) ).
```

```
/* =====  
=====
```

/* =====
=====

AVALIAÇÃO, ACESSO À BASE DE DADOS, E PRODUÇÃO
DE EXPLICAÇÕES E RESULTADOS

```
/* ORBI.PL */
```

```
/******  
/*      LUIS MONIZ PEREIRA, EUGENIO OLIVEIRA, PAUL SABATIER      */  
/*      */  
/*      Nucleo de Inteligencia Artificial                       */  
/*      Departamento de Informatica                            */  
/*      Universidade Nova de Lisboa                            */  
/*      */  
/*      Projecto ORBI - 1981/82                                */  
/******
```

```
/*      Esta e' a versao ?? do programa ORBI.
```

```
Este Programa e' especialista em ordenamento biofisico do territorio */
```

```
/*OPERADORES*/
```

```
?-of(247,xfy,senao).  
?-of(245,xfy,se).  
?-of(243,xfy,ou).  
?-of(241,xfy,e).  
?-of(239,xfy,onde).  
?-of(100,fx,?).  
?-of(35,xfx,!).  
?-of(30,xfy,-).
```

```
/*METACONTROL*/
```

```
?G:- solve(G), assertz(sc(G)),fail.          /* sarbase collection */  
?G:- retract(sc(G)),!,retracore.
```

```
solve(G) :- G,!,
```

```
try(G) :- G, !,  
try(_).
```

```
all_(A,B,C) :- all(A,B,C), achei ; not end, C=[].
```

```
all_(A,B,C) :- B, achei, C=A ; not retract(end), C=[].
```

```
achei :- end, ! ; assert(end).
```

```
/* Lancamento da semantica vindo da gramatica */
```

```
exec( fail ) :- !, fail.
```

```
exec( que(X,Y) ) :- do(Y,X,NX), saida(outnl(NX,3)), fail.
```

```
exec( quant(X,Y) ) :-  
    ( do(Y,X,NX), saida(outnl(NX,3)) ; saida(outnl(40,3)) ), !, fail.
```

```
exec( simnac(L,Y) ) :-  
    ( do(Y,L,L), saida(outrev(wr([e'',verdade])));  
      saida(outrev(wr([nac,e'',verdade])));  
      !, fail.
```

```
do(Y,W,NX) :- ( var(W), X=W ; W=(L=X) );  
              ( explic(D) ; D='n'/L );  
              !,retracore
```

```

    det(Y,O,R,E), n1,n1,
    not (O='n'/_ ,nonvar(C),not saida((outrev(outc(C,X)),n1,n1))), !,
    ( Y\=all_(_,_), nonvar(W),
      !, ( R, ( O='s'/_ , NX=expl_(E) ; NX=X ) ; NX=[] ), ! ;
        ( R=all_(NX,_,_ ) ; NX=X ), !, R ).

```

```

/* compa' , expr, 'e' , 'ou' , 'v' , 'alt' , 'em' , 'max' , 'min' */

```

```

compa(G) :- arg(1,G,X), inteser(X), arg(2,G,Y), inteser(Y), G.

```

```

expr(_,E) :- E.

```

```

A e B :- A , B.

```

```

A ou B :- A ; B.

```

```

v(_,G) :- G.

```

```

alt(NA,NB,EA,EB,E) :- NA, E=EA, ! ; NB, E=EB.

```

```

em(G,H) :- clause(H,B), inc(B,G).

```

```

inc(X,X).

```

```

inc(T,E) :- term(T,A,B), ( inc(A,E) ; inc(B,E) ).

```

```

term(_ onde A, A, nil).

```

```

term(_ se A , A, nil).

```

```

term(A senao B, A, B).

```

```

term((A,B),A,B).

```

```

max([],[]).

```

```

max([[IX,X]IL],M) :- mx(L,X,[[IX]],M), !.

```

```

mx([],_,M,M).

```

```

mx([IE,EDIL],E,I,M) :- mx(L,E,[[IE]],M).

```

```

mx([IE,EDIL],A,I,M) :- inteser(E), A<E, mx(L,E,[[IE]],M).

```

```

mx([IL],A,I,M) :- mx(L,A,I,M).

```

```

min([],[]).

```

```

min([[IX,X]IL],M) :- mn(L,X,[[IX]],M), !.

```

```

mn([],_,M,M).

```

```

mn([IE,EDIL],E,I,M) :- mn(L,E,[[IE]],M).

```

```

mn([IE,EDIL],A,I,M) :- ( not inteser(E) ; A<E ), mn(L,A,I,M).

```

```

mn([IE,EDIL],_,I,M) :- mn(L,E,[[IE]],M).

```

```

/* Fricidade deterministica -- sidetracking */

```

```

det(A se B,O,(NB,A),EB):- det(B,O,NB,EB).

```

```

det(A senao B,O,alt(NA,NB,EA,EB,E),E):- det(A,O,NA,EA) , det(B,O,NB,EB).

```

```

det(A ou B,O,alt(NA,NB,EA,EB,E),E):- det(A,O,NA,EA), det(B,O,NB,EB).

```

```

det(A onde B,O,(NB,A),(A,EB)):- det(B,O,NB,EB).

```

```

det(A e B,O,(A,B),(A,B)).

```

```

det((true,G),O,NG,EG) :- det(G,O,NG,EG).

```

```

det((G,true),O,NG,EG) :- det(G,O,NG,EG).

```

```

det(All_media,D,All_media,true) :- All_media=(al_(_,_,_),media(_,_)).
det((G,GS),D,(NG,NGS),(EG,EGS)) :- det(G,D,NG,EG), det(GS,D,NGS,EGS).
det(all_(X,F,C), D, all_(NX,NF,C), NC) :- det(F,D,NF,E), exf1al(O,X,E,NX,C,NC).
det(al_(X,F,C), D, al_(NX,NF,C), NC) :- det(F,D,NF,E), exf1al(O,X,E,NX,C,NC).
det(not not F,D,NF,E) :- det(F,D,NF,E).
det(not F,D:not NF,true) :- det(F,D,NF,_).
det(d(N,D,V),D,S,NE) :- nonvar(D), des(N,NN),
    C=clause(d(NN,D,V),A), A=(_,tabela(_,_,E)),
    ( nonvar(N), C, S=A ; S=(C,A) ),
    exf1d(O,(d(NN,V),E),NE).
det(expr(N,G),_,G,expr(N,G)).
det(A same B, D, NA same B, EA same B) :- det(A,D,NA,EA).
det(a(A,D,V),D,S,(a(NA,V),E)) :- nonvar(D), des(A,NA),
    C=clause(a(NA,D,V),E),
    ( nonvar(A), C, det(E,D,S,E) ;
      S=(C,solve(det(E,D,NG,E),NG) ).
det(f(F,D,V),D,S,(f(NF,V),E)) :- nonvar(D), des(F,NF),
    C=clause(f(NF,D,V),E),
    ( nonvar(F), C, det(E,D,S,E) ;
      S=(C,solve(det(E,D,NG,E),NG) ).
det(F,_,F,F) :- F=s(_,_,d(.,.,.,.)), asserta(r_(F)), /* essencial d(A) */
det(R,_,R,true) :- R=r(.,.,.,.), asserta(rs_(R)).
det(G,_,E,true) :- Fred(G,I,V-nil,_,NF), des(I,NI),
    functor(E,NF,2),
    ars(1,E,NI), ars(2,E,V).
det(G,_,G,true).
exf1al('n'/_/_,X,_,X,_,true).
exf1al(.,.,E same B,exf1_(E),C,same([E,C])).
exf1al(.,.,same(E),E,C,C).
exf1al(.,.,E,exf1_(E),C.C).
des(G(N,NG),DES) :- nonvar(N), caracteres(N,[],LC),
    conc(LC,[N9],NL), name(DES,NL),
des(N,N).
caracteres(N,LI,[CN|LID]) :- 0=CN, N=CN, CN is 49+N.
caracteres(N,LI,LC) :- N1 is N mod 10+48, N2 is N/10,
    ( N2=0, LC=[N1|LID] ; caracteres(N2,[N1|LID],LC) ).
exf1d('s'/'s',X,X).
exf1d(.,.,true).

```

/* MISCELANEA */

/* calculo de media */


```

media([],0) :- !.
media(R,M) :- soma(R,0,S,0,N), M is (S+N/2)/N, !.

soma([V|VR],S,NS,N,NN) :- integer(V), SI is S+V, NI is N+1,
                           soma(VR,SI,NS,NI,NN).

soma([_|VR],S,NS,N,NN) :- soma(VR,S,NS,N,NN).

soma([],S,S,N,N).

/* construcao de inteiros */

consint([],NI,NI).
consint([C1|CR],I,NI) :- !, C1 = <57, C1> = 48, II is I*10+C1-48, consint(CR,II,NI).

conc([H|T],L,[HIR]) :- conc(T,L,R),
                        conc([],L,L).

isual(X,X) :- !.
isual(0,[nao,etc]) :- !.
isual(1,[pouco,etc]) :- !.
isual(2,[also,etc]) :- !.
isual(3,[etc]) :- !.
isual(4,[bastante,etc]) :- !.
isual(5,[muito,etc]) :- !.

/*
isual(d,'AREAS DEGRADADAS') :- !.
isual(t,'AREAS DE PROTECCAO TOTAL') :- !.
isual(p,'AREAS DE PROTECCAO PARCIAL') :- !.
isual(c,'AREAS COMPLEMENTARES') :- !.
isual(r,'AREAS DE RESERVA DO SUBSOLO').
*/

member(X,[X|_]).
member(X,[_|_]) :- member(X,_).

max(C1,(C2,C3),M) :- max(C2,C3,C4), max(C1,C4,M), !.
max(C1,C2,M) :- ( C1 >= C2, M=C1 ; M=C2 ), !.

/* INPUTS */

car([C]) :- set(C), skip(10), lf((put(C),nl)).

ler(A) :- set(C), palavras(C,F), !, A=F, lf((lf(F),nl)).

/* OUTPUTS */

saida(G) :- lf(G), fail.
saida(G) :- tell(user), G.

lf(G) :- ( imprime, tell('lf:'), assert(lf_),
          ( mlf(G) ; G ), retract(lf_) ; true ), !.

mlf(mess(F)) :- see(F), repeat, set0(C),
               ( eof(C), put(10), seen, ! ; put(C), fail ).

eof(0).
eof(26).

wr([]).

```

```

wr([A|B]) :- wr(A), tab(1), wr(B).      /* wr no MOA.PL */

outnl(X,M) :- out(X,M), !, nl.

out(n:!,_).

out(A=!,M) :- tab(M), outrev(outc(A,E)), nl,nl, N is M+4, outnl(E,N).
out(A;E M) :- tab(M), outc(A,E),
              ( inteser(E), N=4 ; N is M+4,nl ), outl(E,N), nl.
out(A/E,M) :- out(A,M), out(E,1).
out([A],M) :- outl(A,M).
out([A|E],M) :- /* comPr([A|E],L,Flas),
                 ( var(Flas), */          outl([A|E],M)      /* ; outlc(L,M) )*/.
out(A-E,M) :- out(A,M), out(E,3).
out(N,M) :- inteser(N), tab(M), ( N<100, tab(1) ; true ), write(N).
out(expl(E),M) :- expl(E), nl.
out(g(N,NR),M) :- des(g(N,NR),E), tab(M), write(E).
out(arepresentar(R),_) :- nl, arepresentar(R), nl.
out(I,M) :- tab(M), write('hao ha').
out(X,M) :- tab(M), write(X).

/*comPr [[A|As],[A|Es]]L, NL, 1 ) :- ar(As,Es,ARs), comPr([[A|ARs]]L, NL, L).
comPr([[A|As]]L, [[A|As]]NL, Flas ) :- comPr(L,NL,Flas).
comPr(L,[_],_). */

outlc([[A|As]]L, M) :- outnl(A,M), N is M+4, nl,outl(As,M), nl, outlc(L,M).
outlc([],_).

outc(EAD,C) :- out(A^C), vr(A,C).
outc([A|E],[CA|CE]) :- out(A,C), vr(A,[CA]), tab(4), outc(E,CE).
outc(A,E) :- out(A,C), vr(A,[E]).

vr(L,[_]-R) :- R=nil.

vr(valor,_) :- tab(1), write(representatividade).
vr(A,_) :- member(A,[partidao,factor,descriptor]),
           tab(1), write(valor), tab(1), write(representatividade).
vr(_,_).

outl([],M) :- !.

```

```
outl([IHIT]I_L,M) :- outl(IHIT,M), nl, fail.
```

```
outl([L_L]IR,M) :- !, outl(R,M).
```

```
outl([AI_L,M) :- solve((out(A,M), (A=_:_):tab(2)) ), fail.
```

```
outl([L_IB,M) :- !, outl(B,M).
```

```
outl(A,M) :- out(A,M).
```

```
outrev(G) :- imprime, G, ! ; reverse, G, norev.
```

```
/*SAIDA DAS EXPLICACOES*/
```

```
expl((A,_)) :- expl(A), fail.
```

```
expl((_,B)) :- !, expl(B), !.
```

```
expl(a(A,V-R)) :- NV=[_L_], igual(V,NV), tact(A,N),  
                exp(wr([,antidade,A,-,N]),wr(NV),(write(V),tab(1)),R), fg, !.
```

```
expl(f(F,V-R)):- tfact(F,N),  
                ( V=s, W=wr([selectivo]) ; W=write(V) ),  
                exp(wr([,factor,F,-,N]),W,true,R),  
                trs((expli(_/'s'),fg)), !.
```

```
expl(d(D,V-R)):- tdesc(D,N), nl,  
                exp(wr([,descriptor,D,-,N]),write(V),true,R), fg, !.
```

```
expl(p(X,Y,_)):- outnl(ponto=X/Y,4), nl, !.
```

```
expl(mess(A)) :- nl, explm(A), !.
```

```
expl(expr(N,V is Y)) :- ( Y=(W+50)/100, Z=V/100 ; Z=Y ),  
                        write(V), explm(mo16),  
                        write(Z), explm(mo18),  
                        expressao(N,E), write(E), nl,nl, !.
```

```
expl(V is W) :- !.
```

```
expl(write(A)) :- write(A), !.
```

```
expl(_).
```

```
fg:- tab(3), write('forque:'), nl.
```

```
exp(T,W,V,R) :-  
    outrev((T,tab(2),wr([tempo,valor]),W)), nl,  
    put(40), V,  
    wr([com,representatividade]),  
    write(R), tab(1), put(41), tab(1).
```

```
explm(A) :- seeins(F), ( lf_, mlf(mess(A)), ! ; mess(A) ), trimcore, see(F).
```

```
/* Apresentacao das regras ao utilizador */
```

```
apresentar((A:-B)) :- apresentar(A,0), apresentar(B,C), !.
```

```

apresentar((A,B),M):- apresentar(A,M), apresentar(B,M), !.

apresentar(A senao _,M) :- nl, apresentar(A,M), nl, write(senao), nl,nl, fail.
apresentar(_ senao B,M) :- apresentar(B,M), !.

apresentar(A se B,M):- apresentar(A,M), tab(2), write(L is M+4),
                        write(se), nl, apresentar(B,L), !.

apresentar(A ou _,M):- write(N is M+3), apresentar(A,_),
                        nl, tab(M), write(ou), nl, fail.
apresentar(_ ou B,M):- (B=(_ ou _), K=M ; K is M+3), apresentar(B,K), !.

apresentar(A e _,M):- apresentar(A,M), tab(2), write(e), fail.
apresentar(_ e B,_):- apresentar(B,2), !.

apresentar(A onde B,M) :- apresentar(B,M), apresentar(A,M), !.

apresentar(expr(_,X is Y), M) :- tab(M), ( Y=(W+50)/100, Z=W/100 ; Z=Y ),
                                        (X='v.int' ; true),
                                        write(X), write(=), write(Z), nl,nl, !.

apresentar(_ is _,_) :- !.
apresentar(max(_,_,_),_) :- !.
apresentar(B,_):- not inteser(B), pred(B,I,V-,NC,_), constr(I,NC,U), !.
apresentar(true,_) :- !.
apresentar(A,M) :- tab(M), write(A).

pred(a(I,_,VR),I,VR,'a',taft).
pred(f(I,_,VR),I,VR,'f',tfact).
pred(d(I,_,VR),I,VR,'d',tdesc).

constr(X,Y,C):- (nonvar(X), (inteser(X), C1 is X/10, C2 is X mod 10,
                            NC2 is C2+48,
                            (C1=0, LX=[NC2]; NC1 is C1+48, LX=[NC1,NC2])
                            ;
                            name(X,LX)),
                conc(Y,LX,L), name(C,L)
                ;
                var(X), name(C,L), conc(Y,LX,L), name(X,LX)), !.

taft(1,habitacao).
taft(2,'agricultura intensiva').
taft(3,'agricultura nao intensiva').
taft(4,industria).
taft(5,'infraestruturas de recreio').

tfact(1,'conforto climatico').
tfact(2,'capacidade dos terrenos 'a fundacao').
tfact('2a','potencial em aguas-abastecimento').
tfact('2b','resistencia 'a solucao de mantos aquiferos').
tfact('2c','dinamica da vegetacao natural').
tfact(3,'resistencia dos solos 'a erosao').
tfact(4,'resistencia 'a producao agricola').
tfact(5,'resistencia ao incendio').
tfact('6a','riquezas geologicas').
tfact('6b','fitocensores notaveis').
tfact('6c','riquezas culturais').
tfact('6d','paisagens desestabilizadas').
tfact('6e','riquezas do sub-solo').
tfact('6f','riquezas cineceticas e paisagisticas').
tfact(7,'oferta do territorio para o recreio').

```

```

tdesc(1,'qualidade dos microclimas').
tdesc(2,'zonas climaticas homogeneas').
tdesc(3,'permeabilidade do substracto').
tdesc(4,'resistencia mecanica do substracto').
tdesc(5,'estabilidade geodinamica do substracto').
tdesc('5a','inclinacao das encostas').
tdesc('5b','risco potencial de incendio corrido').
tdesc('5c','recursos do sub-solo').
tdesc(6,'sitos e elementos geologicos').
tdesc('6a','arqueologia').
tdesc(7,'escoamento fluvial superficial').
tdesc(8,'potencial de 'asuas subterraneas').
tdesc(9,'vulnerabilidade 'a poluicao dos mantos aquiferos').
tdesc(10,'riscos de erosao').
tdesc('10a','capacidade pedologica').
tdesc(11,'equilibrio da vegetacao').
tdesc(12,'piroresistencia vegetal').
tdesc(13,'riquezas floristicas').
tdesc(20,'potencialidade cinetica e paisagistica').
tdesc('23a','valores historicos construidos').
tdesc('23b','potencial de recreio').
tdesc(24,'areas degradadas').
tdesc(26,'acessibilidade viaria').

```

/* ACESSO 'AS DESCRICOES DOS PONTOS EM DISCO (por Antonio Porto) *

```

r(Xm,XM,Ym,YM) :- nonvar(Xb), !,
    ( retract(rs_( _)),
      base(Xb,Yb),
      slot(Xm,Xb,X), slot(XM,Xb,Xk),
      slot(Ym,Yb,Y), slot(YM,Yb,Yk),
      locate(X,Xk,Y,Yk,Files), !,
      ( var(Files), !,
        indef(mprerr,r(Xm,XM,Ym,YM)) ;
        set_files(Files), asserta(resiac(Xm,XM,Ym,YM)) ) ;
    true ).

```

```

r(Xm,XM,Ym,YM) :- indef(mprind,r(Xm,XM,Ym,YM)).

```

```

p(_,_ ,d(A,_ ,_ ,_ )) :- nonvar(A), !.

```

```

p(X,Y,D) :- var(X), !, pr(X,Y,D).

```

```

p(X,Y,D) :- locate(X,Y,F), F, set_file(F), !, d(X,Y,D).

```

```

p(X,Y,D) :- indef(mprerr,p(_ ,_ ,D)).

```

```

pr(X,Y,D) :- retract(resiac(Xm,XM,Ym,YM)), !,
o(X,Y,D), X <= XM, X >= Xm, Y <= YM, Y >= Ym.

```

```

pr(X,Y,D) :- re_(R), !, R, p(X,Y,D).

```

```

pr(X,Y,D) :- indef(mprind,p(X,Y,D)), !.

```

```

indef(Mes,PR):- saida(mess(Mes)), repeat, saida(mess(mprcf)), morfologia(M,
    testar(M,PR), !, nl, nl, PR.

```

```

testPr([n(X),virs,n(Y)],r(X,Y,_)).

testPr([n(Xm),virs,n(XM),fontoevirs,n(Ym),virs,n(YM)],r(Xm,XM,Ym,YM)).

slot(X,Xb,N) :- Xr is X-Xb, N is Xr-(Xr mod 10)+1.

locate(X,Xk,Y,Yk,Files) :- X <= Xk, locate(Y,Yk,X,Files),
                           X10 is X+10, locate(X10,Xk,Y,Yk,Files), !.
locate(_,_,_,_,_).

locate(Y,Yk,X,Files) :- Y <= Yk, locate(X,Y,File),
                        try((File,member(File,Files))),
                        Y10 is Y+10, locate(Y10,Yk,X,Files).
locate(_,_,_,_).

locate(X,Y,File) :- Xr is X/10, Yr is Y/10,
                   CX1 is Xr/10+48, CX2 is (Xr mod 10)+48,
                   CY1 is Yr/10+48, CY2 is (Yr mod 10)+48,
                   name(File,[114,CX1,CX2,CY1,CY2]).

set_files([]) :- !, fail.
set_files(F1_) :- set_file(F1).
set_files(_Fn) :- set_files(Fn).

set_file(F) :- file_in(F), !.
set_file(F) :- new_file(F), consult(F), !.

new_file(F) :- retract(file_in(_)), asserts(file_in(F)),
              retractall(o(_,_)), trace.

file_in(no).

base(0,0).

/*****>

```

METACONHECIMENTO

```
/* METAC.PL */
```

```
/*QUESTIONARIO METACONHECIMENTO */
```

```
meta:-repeat, mess(m105), ler([R]), (R=#nao;  
      (R=#resnao, #resnao;  
      R=#conceitos, conceitos), nl, reverse,  
      write('Posso abandonar o meta conhecimento, s/n?'),  
      nl, ler([S]), #conceitos, #conceitos, (S=#s; S=#n; fail)), !,  
      nl, write('Quer entrar no modulo de Aquisicao, sim ou nao?'),  
      nl, ler([A]), (A=#sim, #reconsult(aquis), #aquisicao; A=#nao), !,  
      nl, write('Quer consultar o dominio do ORDENAMENTO BIOFISICO, s/n?'),  
      nl, ler([S]), (S=#s; S=#n; so3), !.
```

```
conceitos:-write('Os conceitos fundamentais sao:'), nl, nl,  
      repeat, (tconceitos(X,Y), write(X-Y), nl, fail; true), nl,  
      write('Escreva as letras que lhe interessam separadas por espaço'),  
      nl, ler(C), mess(m107), nl, ler(O),  
      inquirirmeta(C,O), !.
```

```
inquirirmeta([D1|OR], O):-tconceitos(C1,N), nl, nl, write('***'), write(N), nl,  
      constmeta(N,O), inquirirmeta(OR,O).
```

```
inquirirmeta([],_).
```

```
constmeta(N,[D1|OR]):-#resmeta(N,D1), constmeta(N,OR).  
constmeta(_,[]).
```

```
#resmeta(N,O):-functor(F,N,A), #ars(O,F,A), write('*'),  
      (O=2, !, (A, #ars(1,A,X), write(X), nl, fail; true);  
      write(A), nl), !.
```

```
/*METACONHECIMENTO*/
```

conceito(quantidade,instanciacoes,	dependencias,	contribui. para
artidao(5, taft(N,Def,Con),	factores,	sintese_biofisica
factor(16, tfact(F,Def,Con),	descriptorios,	artidoes
descriptor(23, tdesc(D,Def,Con),	pontos-	factores
ponto(25, limcoord(W,Lm,LM),	trabalho_de_campo,	descriptorios
sintese(1, sinbio(S),	artidoes_e_f6a,	metafinal
rede(1, red(R),	factores,	meta(zonas_Laprotectidas)

```
tconceitos(s,sintese).  
tconceitos(a,artidao).  
tconceitos(r,rede).  
tconceitos(f,factor).  
tconceitos(d,descriptor).  
tconceitos(p,ponto).
```

```
red('PRIMEIRA FUNDAMENTAL DA PROTECCAO DA NATUREZA.').  
limcoord(write('variacao das abscissas e das ordenadas'), (1-707,105-709)).
```

```
sinbio('Sintese Biofisica da utilizacao potencial do territorio.').  
sinbio('Sintese Biofisica da utilizacao potencial do territorio',_,_).
```

```
taft(1-'HABITACAO.',_,_).  
taft(2-'AGRICULTURA INTENSIVA.',_,_).  
taft(3-'AGRICULTURA NAO INTENSIVA.',(f6a,f6b,f6c,f2b,f3,f4,f2c),sintese).  
taft(4-'INDUSTRIA.',_,_).  
taft(5-'INFRAESTRUTURAS DE RECREIO.',_,_).
```



```

tfact(1-'conforto climatico.',_,_).
tfact(2-'aptidao dos terrenos 'a fundacao.',_,_).
tfact('2a'-'potencial em aguas-abastecimento.',_,_).
tfact('2b'-'resistencia 'a poluicao de mantos aquiferos.',_,_).
tfact('2c'-'dinamica da vegetacao natural.',_,_).
tfact(3-'resistencia dos solos 'a erosao.',_,_).
tfact(4-'aptidao 'a producao agricola.',_,_).
tfact(5-'resistencia ao incendio.',_,_).
tfact('6a'-'riquezas geologicas.',_,_).
tfact('6b'-'fitocenoses notaveis.',_,_).
tfact('6c'-'riquezas culturais.',_,_).
tfact('6d'-'paisagens desestabilizadas.',_,_).
tfact('6e'-'riquezas do sub-solo.',_,_).
tfact('6f'-'riquezas cineceticas e paisagisticas.',_,_).
tfact(7-'oferta do territorio para o recreio.',_,_).
tfact('7Ba'-'acessibilidade do territorio.',_,_).

```

```

tdesc(1-'qualidade dos microclimas.',_,_).
tdesc(2-'zonas climaticas homogeneas.',_,_).
tdesc(3-'permeabilidade do substracto.',_,_).
tdesc(4-'resistencia mecanica do substracto.',_,_).
tdesc(5-'estabilidade geodinamica do substracto.',_,_).
tdesc('5a'-'inclinacao das encostas.',_,_).
tdesc('5b'-'risco potencial de incendio corrigido.',_,_).
tdesc('5c'-'recursos do sub-solo.',_,_).
tdesc(6-'sitios e elementos geologicos.',_,_).
tdesc('6a'-'arqueologia.',_,_).
tdesc(7-'escoamento pluvial superficial.',_,_).
tdesc(8-'potencial de aguas subterraneas.',_,_).
tdesc(9-'vulnerabilidade 'a poluicao dos mantos aquiferos.',_,_).
tdesc(10-'riscos de erosao.',_,_).
tdesc('10a'-'aptidao pedologica.',_,_).
tdesc(11-'equilibrio da vegetacao.',_,_).
tdesc(12-'firorresistencia vegetal.',_,_).
tdesc(13-'riquezas floristicas.',_,_).
tdesc(20-'potencialidade cinecetica e paisagistica.',_,_).
tdesc('23a'-'valores historicos construidos.',_,_).
tdesc('23b'-'potencial de recreio.',_,_).
tdesc(24-'areas degradadas.',_,_).
tdesc(26-'acessibilidade viaria.',_,_).

```

```

perestrta:-nl,que_lconc(ncr,T),que_regra(T,I),displ(T,I),reverse.!.

```

```

que_lconc(Qa,T):-nl,repeat,
    write('A sua regra permitira calcular conceitos de que tipo? '),
    reverse,write('a'),norev,write('aptidao('),
    reverse,write('f'),norev,write('factor('),
    (Qa=a;nl,ler([T]),nl,((T=a;T=f),!);
    T=d,write('nao se pode mudar'),nl;fail
    Qa=mc,reverse,write('id'),norev,write('escriptor'),nl,ler([T]
    (T=a;T=f;T=d),!).

```

```

que_regra(T,I):-aconceitos(T,Tipo),nl,
    write('Qual o numero de tal '),write(Tipo),nl,
    write('se tiver duvidas chame '),reverse,write('meta'),norev,
    write('conhecimento.'),nl,
    ler([I]),(I=meta:conceitos,!;fail;true),!.

```

```

displ(T,I):-
    (integer(I),CI is I+48,NI=[CI];CI=I,name(CI,NI)),
    (T=d,A=.,[T,I,(DP,DS),_],clause(A,B);
    A=.,[T,I,'B',V-R],page(I,I)).

```

```

concat(LT,NI,LR),/*concat("U",LR,LF),*/
name(U,LR),concat("R",LR,RF),name(R,RF),
clause(A,B)),!,
ecrfx,cabresna(T,I),(T=d,corpo((DF,DS),E) ;
          apresentar((A:-E)) ),!,

corpo((DF,DS),(A,B)):-corpo((DF,DS),A),corpo(E),!.
corpo((DF,DS),ars(X,Y,_)):-
    nl,write('o valor deste descriptor esta' na posicao '
(Y=DF,NX=X;
    Y=DS,NX is X+14).!),
    write(NX),write(' do segundo argumento de cada ponto'),
    nl,write('e a explicacao encontra-se na tabela'),!.
corpo(tabela(N,_,_)):-write(N),!.

cabresna(T,NI):-tconceitos(T,Tipo),write(Tipo),
    functor(F,Tipo,4),F,ars(2,F,A),A=..[A1,NI-Nome,A2,A3],A,
    write(NI-Nome),nl.

apresentar((A:-B)) :- apresentar(A,0), apresentar(B,0).
apresentar((A,B),M):- apresentar(A,M), apresentar(B,M).
apresentar(A senao B,M) :- nl, N is M+2, apresentar(A,N), nl,
    tab(M), write(senao), nl,nl,
    apresentar(B,N).
apresentar(A se B,M):- apresentar(A,0),
    nl, K is M+2,
    tab(K), write(se), nl, apresentar(B,K).
apresentar(A ou B,M):- N is M+3, apresentar(A,N), nl,
    tab(M), write(ou), nl,
    (B=_ ou _), K=M ; K=N), apresentar(B,K).
apresentar(A e B,M):- tab(M), apresentar(A,M), tab(2),
    write(e), apresentar(B,2).
apresentar(A onde B,M) :- apresentar(B,M), apresentar(A,M).
apresentar(expr(L,X is Y),M) :- tab(M),
    (X='valor intermedio' ; true),
    write(X), write(=), write(Y), nl.

apresentar(_ is _,_).
apresentar(max(_,_,_),_).
apresentar(B,_ ) :- not inteser(B), Fred(B,I,V-,ND), constr(I,ND,V).
apresentar(A,M) :- tab(M), write(A).

Fred(a(I,_,VR),I,VR,'a').
Fred(f(I,_,VR),I,VR,'f').
Fred(d(1,_,VR),I,VR,'d').

constr(X,Y,C):-(!nonvar(X)),(inteser(X),C1 is X/10,C2 is X mod 10,
    NC2 is C2+48,
    (C1=0,LX=[NC2];NC1 is C1+48,LX=[NC1,NC2])
    ;
    name(X,LX)),
    concat(Y,LX,L+name(C,L)
    ;
    var(X),name(C,L),concat(Y,LX+L+name(X,LX)),!.

concat([],L,L).
concat([C|R],L,[C|LR]):-concat(R,L,LR).

```

AQUISIÇÃO DE NOVO CONHECIMENTO

```
/* AQUIS.PL */
```

```
/*AQUISICAO*/
```

```
?-consult(baco),consult(vt100),consult(metao).
```

```
aquisicao:-imp, chave, repeat, aqs([ ]),!,
```

```
imp:- write('Quer o protocolo impresso? s/n?'), nl, ler([R]),  
      (R=s, asserta(imprime), !;  
       (R=n, retract(imprime);true),!).
```

```
aqs(L):-persunta(R),que_conc(aq,T),que_regra(T,I),ofcao(R,T,I,L).
```

```
ofcao(R,T,I,L):- (R=n,  
                  lerregra(T,I,L) ;  
                  R=e,displ(T,I),reclmov,nl,nl,lerregra(existe,T,I,L)),!,
```

```
chave:-repeat, erase,pcurs, saida(write('Qual e' a palavra chave?')),  
      ler([C]), raperase,  
      (C=eco,! ; saida(write('Errou! Quer tentar de novo? Sim ou nao?')),  
      ler([R]),(R=sim,fail;R=nao,quest(1,Reg,V))),!,
```

```
persunta(R):-nl,  
      saida(write('A regra a modificar e' Ja' existente ou e' nova?')),  
      nl,ler([R]).
```

```
lerregra(T,NU,L):- saida(write('Nome da regra?')),nl,ler(NOM),nl,  
      readregra(R,T,NU),  
      per(O),  
      (O=sim, numcan, asserta(R),  
       tell(nbaco), saida(write(R)),put(46),  
       nl,tell(user), saida(write('Mais alteracoes s/n?')),nl,  
       ler([S]),  
       (S=s,erase,resetar,aqs([(T;NUM;NOM)|L]));S=n,erase,resetar,  
       fechar([(T;NU;NOM)|L],NC,F),persi(L,NC,F));  
      O=nao, erase,resetar, saida(write('Nova tentativa,s/n?')),  
      nl,ler([X]),(X=s,aqs(L));  
      X=n, resetar,nl,write('adeus.'),!,
```

```
lerregra(existe,T,NU,L):- readregra(R,T,NU),per(O),  
      (O=sim,  
       numcan,  
       tell(nbaco),  
       saida(write(R)), nl,tell(user),  
       saida(write('Mais alteracoes s/n?')),nl,  
       ler([S]),  
       (S=s,resetar,aqs(L));  
       S=n,erase,resetar,  
       fechar(L,NC,F),persi(L,NC,F)  
      ;  
      O=nao,  
       erase,resetar,  
       saida(write('Nova tentativa,s/n?')),nl,  
       ler([X]),(X=s,aqs(L));  
       X=n,nl,write('adeus.'),!  
      ).
```

```
fechar(L,NC,F):- tell(nbaco),told,actualizameta(L,NC,F),  
      (L=[ ];  
      tell(nmeta),  
      write(')('),write(L),write('). '),  
      told).
```

```
per(O):- saida(write('Posso memorizar a nova regra? sim ou nao?')),
```

```

nl,ler([0]),!.

perstresra:-nl,que_conc(MC,T),que_resra(T,I),displ(T,I),reverse,!.

pers1(L,NC,F):- saida(write('Com vista `a actualizacao do metaconhecimento:
outrev(saida(write('s'))),saida(write('im ou ')),
outrev(saida(write('n'))),
saida(write('so.')),nl,
ler([S]),
(S=s,actual(NC,F);S=n),!.

actualizameta([],[],_).
actualizameta([L|LS],[C1|CS],[F1|FS]):- actualizameta(L,C1,F1),
actualizameta(LS,CS,FS).
actualizameta((C1:NU:NOM),NC,F1):- tconceitoE(C1,Conceito),
functor(F,Conceito,4),
F,F=..[CON,Q,I,D,C],
Q1 is Q+1,
F1=..[CON,Q1,I,D,C],
functor(I,X,3),
NC=..[Y,NU-NOM,_,_],
assert(NC).

actual(NC,F):- told, tell(nmetac), act(NC,F), told.

act([],_).
act([NC1|NCR],F):- escolher(F,[NF1|NFR]), act(NC1,NF1), act(NCR,NFR).
act(NC,F):- write(NC),put(46),nl,
(F=0;write(F),assert(F)),!.

escolher([],_).
escolher([F1|FR],[NF1|NFR]):- comparar(F1,FR,NF1), escolher(FR,NFR).

comparar(F,[],F).
comparar(F,[F1|FR],NF):- functor(F,A,4),functor(F1,B,4),
(A=B, NF=0 ;
A<B, comparar(F,FR,NF)),!.

constr(X,Y,C):- (nonvar(X),(integar(X),CX is X+48,LX=[CX];name(X,LX)),
conc(Y:LX,L),name(C,L);
var(X),name(C,L),conc(Y:LX,L),name(X,LX)),!.

conc([],L,L).
conc([C|FD],L,[C|LD]):-conc(R:L,LD).

readresra((CA:-NC),T,NU):- ecrfixo,modelo,ecrmov2,repeat,
saida(write('Escreva a resra com letras minusculas. '),
saida(write(T)),saida(write(NU)),nl,
saida(write('Termine por ponto (.)')),nl,
read(F),
compot R,(CA:-CO),T,NU),limeat(CO,NC).

compot(A:-E),(A:-E),T,NU):- !,vercab(A,T,NU).
compot(A=E,(H:-C),T,NU):- !,verif(A,T,NU),
thead(A,H,V,R,D),
talt(E=V:C,R,D).
compot(A=E se C senao E,(H:-V=E se NC senao NE),T,NU):- !,
verif(A,T,NU),
thead(A,H,V,R,D),
talt(C,NC,R,D),
tbody(E,NE,V,R,D,A).

```

```

compor(A=E se C, (H :- V=E se NC),T,NU) :- verif(A,T,NU),
                                          thead(A,H,V,R,D),
                                          talt(C,NC,R,D).

thead(A,H,V,R,D):- split(A,A1,A2), H=..[A1,A2,D,V-R].

tbody(A senao B,NA senao NB,V,R,D,N) :- tbody(A,NA,V,R,D,N),
                                          tbody(B,NB,V,R,D,N).

tbody(A=E se C, V=E se NC, V,R,D,N) :- thead(A,_,V,R,D), talt(C,NC,R,D).

tbody(NA=X,(expr(N,V is (NX+50)/100),R is (E+50)/100) onde C, V,R,D,A):-
    (NA=A,
     texp(X,NX,E,C,D),
     assexp(N,X/100);
     nl,
     saida(write('D valor da sua resra devia ser nomeado por ')),
     saida(write(A)),!,fail).

tbody(A,NA,V,R,D,N):- talt(A,NA,R,D).

talt(A ou B,NAB,R,D):- talt(A,NA,R,D),
                       talt(B,NB,R,D),
                       mesmou(NA,NB,NAB).

talt(A e B,NAB,R,D):- talt(A,NA,_,D),
                       talt(B,NB,_,D),
                       mesme(NA,NB,R,NAB).

talt(A,(AI,CR) onde C,R,D):- ars(1,A,A1),ars(2,A,A2),
                              functor(A,FA,2),
                              split(A1,A11,A12),
                              ( integer(A2),reconst(FA,A2,AI,U),
                                C=..[A11,A12,D,V-R1], CR=(R=R1)
                              );
                              /*caso da resra ser <COND>=<COND>*/
                              var(A2),
                              reconst(FA,A2,AI,A2),
                              C=..[A11,A12,D,A2-R1],CR=(R=R1)
                              ;
                              atom(A2),split(A2,A21,A22),
                              /*casos de <VAL>=letra ou <ALT> ser <COND>=<COND>*/
                              (A22=true,reconst(FA,A2,AI,U),
                                C=..[A11,A12,D,V-R1],CR=(R=R1)
                              );
                              reconst(FA,U2,AI,U1),
                              C1=..[A11,A12,D,U1-R1],
                              C2=..[A21,A22,D,V2-R2],
                              C=(C1 , C2), CR=max(R1,R2,R)
                              )
                              ).

texp(X1+>R, N1+NXR, R1+RR, (C1,CR), D):- texp(X1,NX1,R1,C1,D),
                                          texp(XR,NXR,RR,CR,D).

texp(X1*X2,Y1*U,X1*R,C,D):- integer(X1), spl(X2,U,R,C,D).

texp(X,NX,F,C,D) :- spl(X,NX,R,C,D).

spl(X,X,1,true,_) :- integer(X).
spl(X,U,F,C,D) :- split(X,X1,X2), C=..[X1,X2,D,V-R].

split(X,X1,X2):-name(Y,(FY|R)),name(X1,(PX1),

```

```
(RX=I,X2=true;
 (consint(RX,0,X2);name(X2,RX))),!,
```

```
reconst(FA,A,NA,U):-NA=..[FA,U,A].
```

```
assexp(N,R):- expressao(N1,_),N is N1+1,asserta(expressao(N,R)).
```

```
mesmo( (A,RA) onde C, (B,RE) onde C, (A,RA) ou (B,RE) onde C),
```

```
mesmo( (A,RA) onde C, ((B1,RB1) onde C) ou ERB,
```

```
((A,RA) ou (B1,RB1) onde C) ou ERB),
```

```
mesmo( NARA , NERB , NARA ou NERB).
```

```
mesme( (A,RA) onde C, (B,RE) onde C, R, ((A e B),max(RA,RE,R)) onde C),
```

```
mesme( (A,RA) onde C, ((B1,RB1) onde C) e (B,RE), R,
```

```
((A e B1),max(RA,RB1,RE,R)) onde C e (B,max(RA,B1,RE,R)) ),
```

```
mesme( (NA,RA) onde CA, (NE,RE) onde CB, R,
```

```
((NA onde CA e NE onde CB), max(RA,RE,R)) ).
```

```
verif(A,T,NU):-name(A,[L1|LF]),name(C,[L1]),(consint(LR,0,N);name(N,LR)),!,
vf(C,N,T,NU).
```

```
vf(C,N,T,NU) :-C=T,N=NU;
```

```
saida( write('o valor da sua regra devia ser nomeado por '))
saida(write(T)), saida(write(NU)),!,fail.
```

```
vercab(A,T,NU) :- pred(A,I,_,C,_), name(NC,C),!,vf(NC,I,T,NU).
```

```
limpar( X, X ) :- var(X).
```

```
limpar(A senao B, LA senao LB) :- limpar(A,LA), limpar(B,LB).
```

```
limpar(A se B, A se LE) :- limpar(B,LE).
```

```
limpar(A ou B, LA ou LB) :- limpar(A,LA), limpar(B,LB).
```

```
limpar(A e B, LA e LE) :- limpar(A,LA), limpar(B,LE).
```

```
limpar(A onde B, LA onde B) :- limpar(A,LA).
```

```
limpar((A,B=B), A).
```

```
limpar((A,max(B,C,D)), (A,LMAX) ) :- limpar(max(B,C,D),LMAX).
```

```
limpar(max(A,X,E), (LMAX,max(LA,D,E)) ) :-
```

```
nonvar(X), X=max(B,C,D),
```

```
limpar(A,LA),
```

```
limpar(max(B,C,D),LMAX).
```

```
limpar(max(A,B,C),max(LA,LE,C)) :- limpar(A,LA), limpar(B,LE).
```

```
limpar(X, A) :- nonvar(X), X=(A=A).
```

```
limpar( X, X).
```

```
modelo:-saida(write('REG:=<COND>=<VAL> se <COR>|EXP|(<CON>=<CON>)'),put(9),
saida(write('IGU:=<COND>=<VAL>'),put(9),put(9),put(9),put(9),nl),
saida(write('COR:=<DIJ> senao <EXP>'),put(9),put(9),put(9),put(9),nl),
saida(write('COND:=<letra minuscula> <ALF>'),
put(9),put(9),put(9),nl),
saida(write('RIJ:=<CJC> ou <CJC>|<CJC>'),
put(9),put(9),put(9),put(9)),
saida(write('ALF:=<numero> numero|<letra minuscula>'),put(9)),
saida(write('CJC:=<IGU> e <IGU>|<IGU>'),
put(9),put(9),put(9),put(9)),
saida(write('VAL:=<numero>'),put(9),put(9),put(9),put(9),nl),
saida(write('EXP:=<CON>=<PAR> + <PAR>|<PAR>'),
put(9),put(9),put(9)),
saida(write('PAR:=<numero> * <CON>'),put(9),put(9),put(9),nl).
```

```

ler(A) :- (set(C), palavras(C, AI), !, AI=A), !, lp(wr(AI)). /*tirei o ~*/
wr([]).
wr([A|B]) :- write(A), tab(1), wr(B).

palavras(46, []). /*ponto*/
palavras(10, []). /*line feed*/
palavras(C, Ps) :- espacador(C), set(NC), palavras(NC, Ps).
palavras(C, [P|Ps]) :- palavra(C, NC, CP), (consint(CP, 0, P) & name(P, CP)),
    palavras(NC, Ps).

palavra(C, C, []). /*espacador(C)*/
palavra(C1, NC, [C1|Cn]) :- set0(C2), palavra(C2, NC, Cn).

espacador(32). /*espaco*/
espacador(44). /*virgula*/
espacador(59). /*ponto e virgula*/
espacador(46). /*ponto*/
espacador(10). /*line feed*/

consint([C1|CR], I, NI) :- !, C1=<57, C1>=48, II is I*10+C1-48, consint(CR, II, NI).
consint([], NI, NI).

/*METACONTROL*/

^G :- solve(G), assertz(sc(G)), fail.
^G :- retract(sc(G)), trimcore.

solve(G) :- G, !.

saida(G) :- lp(G), fail.
saida(G) :- G.

lp(G) :- (imprime, tell('lp:'),
    (mlp(G) ; solve(G)),
    tell(user)
    ;
    true), !.

mlp(mess(F)) :- see(F), repeat, set0(C),
    (C=26, seen, ! ; put(C), fail).

outrev(G) :- imprime, G, ! ; reverse, G, noev.

/* Apresentacao das regras ao utilizador */

apresentar((A:-B)) :- apresentar(A, 0), apresentar(B, 0), !.

apresentar((A,B), M) :- apresentar(A, M), apresentar(B, M), !.

apresentar(A senao B, M) :- nl, apresentar(A, M), nl, write(senao), nl, !, fail.
apresentar(_ senao B, M) :- apresentar(B, M), !.

apresentar(A se B, M) :- apresentar(A, M), tab(2), L is M+4,
    write(se), nl, apresentar(B, L), !.

apresentar(A ou B, M) :- N is M+3, apresentar(A, N),
    nl, tab(M), write(ou), nl, fail.

apresentar(_ ou B, M) :- (B=(_ ou _), K=M ; K is M+3), apresentar(B, K), !.

```



```

apresentar(A e B,M):- apresentar(A,M), tab(2), write(e), fail.
apresentar(_ e B,_):- apresentar(B,2), !.

apresentar(A onde B,M) :- apresentar(B,M), apresentar(A,M), !.

apresentar(lex(r(L),X is Y), M) :- tab(M), ( Y=(W+50)/100, Z=W/100 ; Z=Y ),
      (X='v.int' ; true),
      write(X), write(=), write(Z), nl, nl, !.

apresentar(_ is _,_) :- !.
apresentar(max(_,_,_),_) :- !.
apresentar(B,_) :- not integer(B), pred(B,I,V_1,NC,_), constr(I:NC,V), !.
apresentar(true,_) :- !.
apresentar(A,M) :- tab(M), write(A).

```

```

pred(a(I,_,VR),I,VR,'a',taft).
pred(f(I,_,VR),I,VR,'f',tfact).
pred(d(I,_,VR),I,VR,'d',tdesc).

```

```

constr(X,Y,C):- (nonvar(X), (integer(X), C1 is X/10, C2 is X mod 10,
      NC2 is C2+48,
      (C1=0, LX=[NC2]; NC1 is C1+48, LX=[NC1,NC2])
      ;
      name(X,LX)),
      conc(Y,LX,L), name(C,L)
      ;
      var(X), name(C,L), conc(Y,LX,L), name(X,LX)), !.

```

INTERROGAÇÃO POR MENU

```
/* MENU.PL */
```

```
?-op(247,xfy,se).
?-op(245,xfy,senao).
?-op(243,xfy,ou).
?-op(241,xfy,e).
?-op(239,yfx,onde).
?-op(238,fx,com).
?-op(100,fx,^).
?-op(35,xfx,;).
?-op(30,xfy,-).
```

```
so2 :- /*ciclo principal*/
```

```
    imp, meta, quest(I,Res,V),!,
    ( member(ouve,[Res,I,V]),!,fail ; true ),
    ( integer(I),Pontos(Res,V,I);I=var,artidoes(Res);
      I=outras,persuntas(Res)),!,
    so3.
```

```
imp:- mess(mm1), nl, car(R),
    ( R='s', assert(imp:imir), ! ;
      ( R='n', retract(imp:imir) ; true), !).
```

```
so3 :- see(user), continuar(C), (C=nao,mess(mm2),halt; C=sim,fail).
```

```
/*QUESTIONARIO CONHECIMENTO */
```

```
quest(I,Res,NV):- repeat,erase,
    mess(mm3),reverse,write('n'),
    notev,mess(mm4),
    reverse,write('a'),notev,mess(mm5),
    set(Q),(Q=a,asserta(1([I])));
      Q=n,reconsult(inbaco),erase),!,
    mess(m44), car(Q),
    (Q='o', !, fail ; obter(Q,I,Res,NV), !).
```

```
comess:-l(L),(L=[];nl,mess(mm6),comamess(L)).
```

```
comamess([]).
```

```
comamess([C:NU:NO]):-nl,(C=a,write(NU),write('-'),write(NO);true),nl,
comamess([A,R]):-comamess([A]),comamess([R]).
```

```
obter(artidoes,var,Res,var) :- repeat,nl,mess(m45),
    car(Res),
    (Res='o' ; Res='f' ; Res='n').
```

```
obter(Pontos,I,Res,NV) :-
    repeat,mess(m47),
    car(Res),
    ( Res='o' ;
      ( Res='t' ; Res='r' ),
      repeat,mess(m48),mess(m100),
      ler(I), ( I=ouve ;
        I=outras,NV=true;
        integer(I),
        repeat,mess(m49),
        ler(NV),
        (NV=ouve ;
          member(NV,[Inuito,arto],[bastante,arto],[algo,arto],
            [arto],[pouco,arto],[nao,arto])) ),!,!).
```

```
/*PESQUISA DOS VALORES DAS ARTIDOES*/
```

```
artidoes(resiao):-
```

```
coord(Xmin,Xmax,Ymin,Ymax), mess(m92),
car(M),
```

```
( M='m',
  repeat, mess(m48), comss,
  nl: ler(J), solve((J=[N,NQ], I=a(N,[NQ]) ; [I]=J )),
  ( !ouve,!, fail ; true ),
  (all( V,
    det_((Ponto(X,Y,D), resiao(Xmin,Xmax,Ymin,Ymax,X,Y),
      a(I,D,U)), 0, S, _),!, S),
    R1 ), media(R1,Media),!,
  outmedia(Xmin,Xmax,Ymin,Ymax,I,Media) ;
  mess(m43),!, fail )
  ;
```

```
M='t',
  (all( X/Y:([numero,valor,representatividade]=RP),
    (Ponto(X,Y,D), resiao(Xmin,Xmax,Ymin,Ymax,X,Y),
      det(all_(I/V,a(I,D,U),RP),S,E,0),
        S),
      R);
  mess(m43),!, fail ),
  outgeral(0,[Pontos,artidos]=R,R);
```

```
M='u',
  mess(m48), comss,
  ler([I]), inteser(I),
  det( (Ponto(X,Y,D), resiao(Xmin,Xmax,Ymin,Ymax,X,Y),
    a(I,D,U)), S,E,0),
  naofalha((S,outgeral(0,[Ponto,artidao]=(X/Y:I/V),E)), mess(m43));
```

```
M='o',
  !, fail ),
```

```
artidos(Ponto):- repeat, mess(m48),
  comss,
  ler(J),
  solve(( J=[N,NQ], I=a(N,[NQ]) ; J=[todas] ; [I]=J )),
  mess(m90),
  mess(m47),
  morfologia(Par,_),
  ( Par=[ouve],!, fail ; Par=[X,vins,Y] ),
  inteser(X), inteser(Y),!,
  det( (P(X,Y,D), a(I,D,U)), S,E,0),
  naofalha((S,outgeral(0,[Ponto,artidao]=(X/Y:I/V),E)), true);
```

```
/* PESQUISA DAS COORDENADAS DOS PONTOS QUE SATISFAZEM UMA CONDICAO */
```

```
Pontos(resiao,NV,I) :- igual(V,NV), coord(Xmin,Xmax,Ymin,Ymax),
  det( ( Ponto(X,Y,D), resiao(Xmin,Xmax,Ymin,Ymax,X,Y),
    a(I,D,U-_) ),
    S,E,0),
  naofalha((S,outgeral(0,Ponto=(X/Y),E)), (mess(m46),write(I),nl)).
```

```
Pontos(todos,NV,I) :- igual(V,NV), det( ( P(X,Y,D), a(I,D,U-_) ),
  S,E,0),
  naofalha((S,outgeral(0,Ponto=(X/Y),E)),
  (mess(m46),write(I),nl)).
```

```
coord(Xmin,Xmax,Ymin,Ymax) :- repeat, mess(m91), mess(m47),
  morfologia(C,_),
  ( C=[ouve],!, fail ;
```

```

C=[Xmin,virs,Xmax,pontoevirs,Ymin,virs,Ymax],
inteser(Xmin),inteser(Xmax),
inteser(Ymin),inteser(Ymax) ).

```

```

/*OBTENCAO DE PONTOS COM CARACTERISTICAS ANALOGAS DE VALORES DE APTIDÕES
E/OU FACTORES E/OU DESCRITORES*/

```

```

persuntas(Res):- mess(m101), nl, car(A),
( A='n', ! ; mess(m102), nl, ler(VA) ),
isual(VA,NVA), nl, mess(m103), nl, car(F),
( F='n', ! ; mess(m102),nl, ler([VF])),
nl, mess(m104), nl, car(DE),
( DE='n', ! ; mess(m102), nl, ler([VD]) ),
nl, constrans(A,NVA,F,VF,DE,VD,S),
pesquisa(Res,S).

constrans(A,VA,F,VF,DE,VD,S) :- ( A='n', SA=nao, ! ; consap(A,VA,D,SA) ),
( F='n', SF=nao, ! ; consfac(F,VF,D,SF) ),
( DE='n', SD=nao, ! ; consdes(DE,VD,D,SD) ),
S=(D,SD,SF,SA).

pesquisa(todos,(D,SA,SF,SD)):- det( (P(X,Y,D),SD,SF,SA),S,E,O),
naofalha((S,outseral(D,ponto=(X/Y),E)),
mess(m46)),nl.

pesquisa(resiao,(D,SA,SF,SD)):-coord(Xmin,Xmax,Ymin,Ymax),
det( (ponto(X,Y,D),
resiao(Xmin,Xmax,Ymin,Ymax,X,Y),
SD,SF,SA), S,E,O),
naofalha((S,outseral(D,ponto=(X/Y),E)),
mess(m106)),nl.

consap(AU,NVAU,D,SA):-isual(VAU,NVAU),SA=a(AU,D,VAU-C).

consap((A1,AR),(NVA1,NVAR),D,SA):-isual(VA1,NVA1),SA=(a(A1,D,VA1-C).SAR),
consap(AR,NVAR,D,SAR).

consfac(FU,VFU,D,f(FU,D,VFU-C)).

consfac((F1,FR),(VF1,VFR),D,SF):-SF=(f(F1,D,VF1-C),SFR),
consfac(FR,VFR,D,SFR).

consdes(DU,VDU,D,d(DU,D,VDU-C)).

consdes((D1,DR),(VD1,VDR),D,SD):-SD=(d(D1,D,VD1-C),SDR),
consdes(DR,VDR,D,SDR).

outmedia(Xmin,Xmax,Ymin,Ymax,I,H) :-
isual(H,NH),
mess(m03),
write(I),
mess(m04),
write(Xmin-Ymax),
mess(m05),
write(Ymin-Ymax),
mess(m06),
reverse,wr(NH),ndrev,nl.

naofalha(G,H):- G,encontrei, fail.

```

```
naofalha(_,M):- notenc,M.  
naofalha(_,_).
```

```
encontrei:- enc,!.  
encontrei:- assert(enc).
```

```
notenc:- not retract(enc).  
continuar(C) :- nl, mess(mo13),  
                mess(mo17), car([C]), nl.
```

```
explicacao(media).
```

BASE DE CONHECIMENTOS

```
/* BACO.PL */
```

```
/*OPERADORES*/
```

```
?-op(247,xfy,senao).  
?-op(245,xfy,se).  
?-op(243,xfy,ou).  
?-op(241,xfy,e).  
?-op(239,xfy,onde).  
?-op(10,fx,^).  
?-op(35,fx,!).  
?-op(30,xfy,-).
```

```
/*REGRAS DE INFERENCIA*/
```

```
/*APTIDOES*/
```

```
a(1,D,U-R):-
```

```
U=0 se
```

```
(F4=4 ou F4=5) onde f(4,D,F4-R)  
ou (F6A=s ou F6A=3) onde f('6a',D,F6A-R)  
ou (F6B=0 e F6B=e e F6B=f) onde f('6b',D,F6B-R)  
ou F6C=5 onde f('6c',D,F6C-R)  
ou F6F=5 onde f('6f',D,F6F-R)
```

```
senao
```

```
(exat(6,U is (40*F1+25*F2+20*F2A+5*F2B+5*F3+5*F5+50)/100);  
R is (40*R1+25*R2+20*R3+5*R4+5*R5+5*R6+50)/100)  
onde (f(1,D,F1-R1),f(2,D,F2-R2),f('2a',D,F2A-R3),  
f('2b',D,F2B-R4),f(3,D,F3-R5),f(5,D,F5-R6)).
```

```
a(2,D,V-R):-
```

```
V=0 se
```

```
F6A=s onde f('6a',D,F6A-R)  
ou (F6B=0 e F6B=f) onde f('6b',D,F6B-R)  
ou F6C=5 onde f('6c',D,F6C-R)
```

```
senao
```

```
(exat(7,V is (50*F4+25*F2A+10*F3+10*(6-F2D)+5*F2B+50)/100);  
R is (50*R1+25*R2+10*R3+10*R4+5*R5+50)/100)  
onde (f(4,D,F4-R1),f('2a',D,F2A-R2),f(3,D,F3-R3),  
f('2c',D,F2C-R4),f('2b',D,F2B-R5)).
```

```
a(3,D,V-R):- /* nao esquecer de mudar na base de dados inf em '6' */
```

```
V=0 se
```

```
V4=s onde f('6a',D,V4-R)  
ou V5=s onde f('6b',D,V5-R)  
ou V6=s onde f('6c',D,V6-R)
```

```
senao
```

```
(exat(8,V is (25*(6-F2D) + 10*F3 + 5*F2B + 60*F4 + 50)/100);  
R is (25*R4+10*R2+5*R1+60*R3+50)/100)  
onde  
(f('2b',D,F2B-R1),f(3,D,F3-R2),f(4,D,F4-R3),  
f('2c',D,F2C-R4)).
```

```
a(4,D,V-R):-
```

```
V=0 se
```

```
(F4=4 ou F4=5) onde f(4,D,F4-R)
```



```

ou (F6A=s ou F6A=3) onde f('6a',D,F6A-R)
ou (F6B\=0 e F6B\=e e F6B\=f) onde f('6b',D,F6B-R)
ou F6C=5 onde f('6c',D,F6C-R)
ou F6F=5 onde f('6f',D,F6F-R)
senao
((expr(9,VI is 30*F2+30*F2B+15*F2A+15*F3+10*F5),
  RI is (30*R1+30*R2+15*R3+15*R4+10*R5))
onde (f(1,D,F2-R1),f('2b',D,F2B-R2),f('2a',D,F2A-R3),
  f(3,D,F3-R4),f(5,D,F5-R5)),
((F6E=0 e expr(13,V is (VI+50)/100), R is (RI+50)/100)
ou (F6E=z e expr(10,V is (VI*3/4)+50/100),
  R is (RI*R6+50)/100))
onde f('6e',D,F6E-R6)).

```

```

a(5,D,V-R):-

```

```

  V=0 se

```

```

    (F4=4 ou F4=5) onde f(4,D,F4-R)
    ou (F6A=s ou F6A=3) onde f('6a',D,F6A-R)
    ou F6C=5 onde f('6c',D,F6C-R)

```

```

  senao

```

```

    ((expr(11,VI is 70*F7+20*F1+10*F2),
      RI is 70*R1+20*R2+10*R3)
    onde (f(7,D,F7-R1),f(1,D,F1-R2),f(2,D,F2-R3)),

```

```

    expr(12,V is (VI*9/10+50)/100) se

```

```

      (F6F=5 , R is (RI+50)/100) onde f('6f',D,F6F-R)
      ou (F6B\=0 e F6B\=e e F6B\=f ,
        R is (RI+50)/100) onde f('6b',D,F6B-R)

```

```

      senao

```

```

        (expr(14,V is (VI+50)/100),
          R is (RI+50)/100) ).

```

```

r(rede,D,V-R):- (V=d se F6D=a onde f('6d',D,F6D-R))

```

```

  senao

```

```

    (V=t se F6A=s onde f('6a',D,F6A-R)
    ou (F6B=5 ou F6B=a ou F6B=b) onde f('6b',D,F6B-R)
    ou F6C=5 onde f('6c',D,F6C-R)
    ou F4=5 onde f(4,D,F4-R))

```

```

  senao

```

```

    (V=f se F6A=3 onde f('6a',D,F6A-R)
    ou (F6B=c ou F6B=o) onde f('6b',D,F6B-R)
    ou f4=5 onde f(4,D,F4-R))

```

```

  senao

```

```

    (V=c se F6B=g onde f('6b',D,F6B-R)
    ou F6F=5 onde f('6f',D,F6F-R))

```

```

  senao

```

```

    (V=r se F6E=a onde f('6e',D,F6E-R)).

```

```

/*FACTORES*/

```

```

f(1,D,V-R):-expr(1,VI is (60*D1+40*D2)/10) onde (d(1,D,D1-R1),
  b(2,D,D2-R2)),

```

```

  R is (60*R1+40*R2+50)/100,

```

```

  (VI = <25 e V=1 ou

```

```

  VI > 25 e VI = <33 e V=2 ou

```

```

  VI > 33 e VI = <39 e V=3 ou

```

```

  VI > 39 e VI = <45 e V=4 ou

```

VI>45 e V=5).

f(2,D,V-R):- (expr(2,V is (65*D4+15*D5A+10*D3+10*D5+50)/100),
R is (65*R1+15*R2+10*R3+10*R4+50)/100)
onde (d(4,D,D4-R1),d('5a',D,D5A-R2),d(3,D,D3-R3),
d(5,D,D5-R4)).

f('2a',D,V-R):- V=U onde d(8,D,U-R).

f('2b',D,V-C):- V=U onde d(9,D,U-C).

f('2c',D,V-R):- V=U onde d(11,D,U-R).

f(3,D,V-R):- (expr(3,V is (30*(6-D7)+70*D10+50)/100),
R is (30*C1+70*C2+50)/100)
onde (d(7,D,D7-C1),d(10,D,D10-C2)).

f(4,D,V-C):- V=U onde d('10a',D,U-C).

f(5,D,V-R):- (expr(4,V is (85*D12+15*D5B+50)/100),
R is (85*R1+15*R2+50)/100)
onde (d(12,D,D12-R1),d('5b',D,D5B-R2)).

f('6a',D,V-C1):-
(V=s se V1\=0
e V1\=2
e V1\=3
senao
V=V1) onde d(6,D,V1-C1).

f('6b',D,F6B-R):-
(F6B=s se
(D13=e ou D13=f)
e D11=3
senao
(F6B=0 se
D13=0 ou D11=0)
ou F6B=D13)) onde (d(11,D,D11-R1),d(13,D,D13-R2)),
max(R1,R2,R).

f('6c',D,V-R):-
V=0 se
((V1=0, R=C1)
ou (V2=0, R=C2)) onde (d('6a',D,V1-C1),d('6b',D,V2-C2))
senao
(V=s e R=C5)

f('6d',D,V-R):- V=U onde d(24,D,U-R).

f('6e',D,V-R):-
V=a se
(D11=3 e D5C=a) onde (d(11,D,D11-R1),d('5c',D,D5C-R2))
e max(R1,R2,R)
senao
(V=0 e R=C5)

f('6f',D,V-R):-
V=5 se
(D20=5 ou B20=4) onde d(20,D,D20-R)
senao
(V=0 e R=C5)

f(7,D,V-R):- (expr(5,V is (60*D23B+40*B20+50)/100),
R is (60*R1+40*R2+50)/100)

onde (d(20,D,D20-R1),d('23b',D,D23E-R2)).

f('8a',D,U-R):- V=U onde d(26,D,U-R).

/*DESCRIPTORES*/

d(1,D,VR) :-	args(1,1,2,D,VR),	tabela(1,VR,E).
d(2,D,VR) :-	args(1,3,4,D,VR),	tabela(2,VR,E).
d(3,D,VR) :-	args(1,5,6,D,VR),	tabela(3,VR,E).
d(4,D,VR) :-	args(1,7,8,D,VR),	tabela(4,VR,E).
d(5,D,VR) :-	args(1,9,10,D,VR),	tabela(5,VR,E).
d('5a',D,VR) :-	args(1,11,12,D,VR),	tabela('5a',VR,E).
d('5b',D,VR) :-	args(2,1,2,D,VR),	tabela('5b',VR,E).
d('5c',D,VR) :-	args(2,3,4,D,VR),	tabela('5c',VR,E).
d(6,D,VR) :-	args(2,5,6,D,VR),	tabela(6,VR,E).
d('6a',D,VR) :-	args(2,7,8,D,VR),	tabela('6a',VR,E).
d(7,D,VR) :-	args(2,9,10,D,VR),	tabela(7,VR,E).
d(8,D,VR) :-	args(2,11,12,D,VR),	tabela(8,VR,E).
d(9,D,VR) :-	args(3,1,2,D,VR),	tabela(9,VR,E).
d(10,D,VR) :-	args(3,3,4,D,VR),	tabela(10,VR,E).
d('10a',D,VR) :-	args(3,5,6,D,VR),	tabela('10a',VR,E).
d(11,D,VR) :-	args(3,7,8,D,VR),	tabela(11,VR,E).
d(12,D,VR) :-	args(3,9,10,D,VR),	tabela(12,VR,E).
d(13,D,VR) :-	args(3,11,12,D,VR),	tabela(13,VR,E).
d(20,D,VR) :-	args(4,1,2,D,VR),	tabela(20,VR,E).
d('23a',D,VR) :-	args(4,3,4,D,VR),	tabela('23a',VR,E).
d('23b',D,VR) :-	args(4,5,6,D,VR),	tabela('23b',VR,E).
d(24,D,VR) :-	args(4,7,8,D,VR),	tabela(24,VR,E).
d(26,D,VR) :-	args(4,9,10,D,VR),	tabela(26,VR,E).

args(N,A1,A2,D,U-R) :-
(D=d(A,_,_,_), var(A), F_(F), F=F(.,.,D), !, F ; true),
ans(N,D,Dn), ans(A1,Dn,U), ans(A2,Dn,R).

/* EXPRESSOES */

expressao(14,(v.int//100)).
expressao(13,(v.int//100)).
expressao(12,(9/10 do v.anterior//100)).
expressao(11,(70*f7+20*f1+10*f2)).
expressao(10,(3/4 do v.anterior)).
expressao(9,(3*(f2+30*f2b+15*f2a+15*f3+10*f5)).
expressao(8,(25*(6-f2c)+10*f3+5*f2b+60*f4)/100).
expressao(7,(50*f4+25*f2a+10*f3+10*(6-f2c)+5*f2b)/100).
expressao(6,(40*f1+25*f2+20*f2a+5*f2b+5*f3+5*f5)/100).
expressao(5,(60*d23b+40*d20)/100).
expressao(4,(15*d12+15*d5b)/100).
expressao(3,(10*(6-d7)+70*d10)/100).
expressao(2,(65*d4+15*d5a+10*d3+10*d5)/100).
expressao(1,(60*d1+40*d2)/10).

/*TABELAS INDICADORAS DE MENSAGENS*/

tabela(1,U-L,(mess(mD1),mess(X))):- t1(U,X), !.

```

t1(1,md1a).
t1(2,md1b).
t1(3,md1c).
t1(4,md1d).
t1(5,md1e).

tabela(2,U-_,(mess(md2),write(X),mess(md2a,))):-t2(U,X),!.

t2(1,'MUITO POUCO').
t2(2,'POUCO').
t2(3,'MEDIANAMENTE').
t2(4,'').

tabela(3,U-_,(mess(md3),mess(X))):- t3(U,X),!.

t3(1,md3a).
t3(2,md3b).
t3(3,md3c).
t3(4,md3d).
t3(5,md3e).

tabela(4,U-_,(mess(md4),mess(X))):- t4(U,X),!.

t4(1,md4a).
t4(2,md4b).
t4(3,md4c).
t4(4,md4d).
t4(5,md4e).

tabela(5,U-_,(mess(md5),X)):-t5(U,X),!.

t5(3,mess(md5c)). /* 1,2,4 nao ocorrem mesmo ? */
t5(5,mess(md5e)).

tabela(9,U-_,mess(X)) :- t9(U,X),!.

t9(0,m1).
t9(1,m2).
t9(2,m3).
t9(3,m4).
t9(4,m5).
t9(5,m6).

tabela(7,U-_,E) :- t7(U,E),!.

t7(0,mess(m7)).
t7(U,(mess(m7a),write(Y),mess(m7b),write(X),nl,nl)) :- tL7(U,X,Y).

tL7(1,'INFERIOR A 150' , 'MUITO BAIXO').
tL7(2,'ENTRE 150 E 400' , 'BAIXO').
tL7(3,'ENTRE 400 E 1000' , 'MEDIO').
tL7(4,'ENTRE 1000 E 1500' , 'ELEVADO').
tL7(5,'SUPERIOR A 1500' , 'MUITO ELEVADO').

tabela(6,U-_,E) :- tL6(U,E),!.

t6(0,mess(m9)).
t6(2,mess(m10)).
t6(3,mess(m11)).
t6(U,(mess(m12),mess(X))):- tL6(U,X).

tL6(a,m53).
tL6(b,m54).
tL6(c,m55).
tL6(d,m56).

```

```
t1(1,md1a).
t1(2,md1b).
t1(3,md1c).
t1(4,md1d).
t1(5,md1e).
```

```
tabela(2,V-L,(mess(md2),write(X),mess(md2a))):-t2(V,X),!.
```

```
t2(1,'MUITO POUCO').
t2(2,'POUCO').
t2(3,'MEDIANAMENTE').
t2(4,'').
```

```
tabela(3,V-L,(mess(md3),mess(X))):-t3(V,X),!.
```

```
t3(1,md3a).
t3(2,md3b).
t3(3,md3c).
t3(4,md3d).
t3(5,md3e).
```

```
tabela(4,V-L,(mess(md4),mess(X))):-t4(V,X),!.
```

```
t4(1,md4a).
t4(2,md4b).
t4(3,md4c).
t4(4,md4d).
t4(5,md4e).
```

```
tabela(5,V-L,(mess(md5),X)):-t5(V,X),!.
```

```
t5(3,mess(md5c)), /* 1,2,4 nao ocorrem mesmo ? */
t5(5,mess(md5e)).
```

```
tabela(9,V-L,mess(X)) :- t9(V,X),!.
```

```
t9(0,m1).
t9(1,m2).
t9(2,m3).
t9(3,m4).
t9(4,m5).
t9(5,m6).
```

```
tabela(7,V-L,E) :- t7(V,E),!.
```

```
t7(0,mess(m7)).
t7(V,(mess(m7a),write(Y),mess(m7b),write(X),nl,nl)) :- tL7(V,X,Y).
```

```
tL7(1,'INFERIOR A 150' , 'MUITO BAIXO').
tL7(2,'ENTRE 150 E 400' , 'BAIXO').
tL7(3,'ENTRE 400 E 1000' , 'MEDIO').
tL7(4,'ENTRE 1000 E 1500' , 'ELEVADO').
tL7(5,'SUPERIOR A 1500' , 'MUITO ELEVADO').
```

```
tabela(6,V-L,E) :- t6(V,E),!.
```

```
t6(0,mess(m9)).
t6(2,mess(m10)).
t6(3,mess(m11)).
t6(V,(mess(m12),mess(X)) :- tL6(V,X).
```

```
tL6(a,m53).
tL6(b,m54).
tL6(c,m55).
tL6(d,m56).
```

```
t_6(e,m57).
t_6(f,m58).
t_6(g,m59).
t_6(h,m60).
t_6(i,m61).
t_6(j,m62).
t_6(k,m63).
```

```
tabela(13,U_,mess(E)) :- t13(U,E),!.
```

```
t13(0,m13).
t13(f,m14).
t13(e,m15).
t13(d,m16).
t13(c,m17).
t13(b,m18).
t13(a,m19).
t13(5,m20).
```

```
tabela(20,U_,E) :- t20(U,E),!.
```

```
t20(0,mess(m21)).
t20(U,(mess(m22),write(X),nl,nl)) :- t_20(U,X).
```

```
t_20(1,'MUITO BAIXA').
t_20(2,'BAIXA').
t_20(3,'MEDIA').
t_20(4,'ELEVADA').
t_20(5,'MUITO ELEVADA').
```

```
tabela('6a',U_,E) :- t6a(U,E),!.
```

```
t6a(0,mess(m23)).
t6a(U,(mess(m24),mess(X))) :- t_6a(U,X).
```

```
t_6a(a,m64).
t_6a(b,m65).
t_6a(c,m66).
t_6a(d,m67).
t_6a(e,m68).
t_6a(f,m69).
t_6a(g,m70).
t_6a(h,m71).
t_6a(i,m72).
t_6a(j,m73).
```

```
tabela('23a',U_,E) :- t23a(U,E),!.
```

```
t23a(0,mess(m25)).
t23a(U,(write('Ocorre '),mess(X),mess(m26))) :- t_23a(U,X).
```

```
t_23a(a,m74).
t_23a(b,m75).
t_23a(c,m76).
t_23a(d,m77).
t_23a(e,m78).
t_23a(f,m79).
t_23a(g,m80).
t_23a(h,m81).
t_23a(i,m82).
t_23a(j,m83).
```

```
tabela(11,U_,E) :- t11(U,E),!.
```

```
t11(n,mess(m27)).
```

```
t11(V,(mess(m28a),write(X),mess(m28b),mess(Y))) :- t_11(V,X,Y).
```

```
t_11(1,'MUITO BAIXO',m85).  
t_11(2,'BAIXO',m86).  
t_11(3,'MEDIO',m87).  
t_11(4,'MUITO ELEVADO',m88).  
t_11(5,'MUITO ELEVADO',m89).
```

```
tabela(10,V_-,mess(E)) :- t10(V,E),!.
```

```
t10(0,m29).  
t10(1,m30).  
t10(2,m31).  
t10(3,m32).  
t10(4,m33).  
t10(5,m34).
```

```
tabela('10a',V_-,mess(E)) :- t10a(V,E),!.
```

```
t10a(1,md10a1).  
t10a(2,md10a2).  
t10a(3,md10a3).  
t10a(4,md10a4).  
t10a(5,md10a5).
```

```
tabela('5a',V_-,mess(md5a),mess(X)) :- t5a(V,X),!.
```

```
t5a(1,md5a1).  
t5a(2,md5a2).  
t5a(3,md5a3).  
t5a(4,md5a4).  
t5a(5,md5a5).
```

```
tabela('5b',V_-,mess(X)) :- t5b(V,X),!.
```

```
t5b(1,md5b1).  
t5b(5,md5b5).
```

```
tabela('5c',V_-,mess(X)) :- t5c(V,X),!.
```

```
t5c(a,md5ca).  
t5c(0,md5c0).
```

```
tabela(8,V_-,mess(MD8)) :- t8(V,MD8),!.
```

```
t8(1,md81).  
t8(2,md82).  
t8(3,md83).  
t8(4,md84).  
t8(5,md85).
```

```
tabela(12,V_-,mess(MD12)) :- t12(V,MD12),!.
```

```
t12(1,md121).  
t12(2,md122).  
t12(3,md123).  
t12(4,md124).  
t12(5,md125).
```

```
tabela('23b',V_-,mess(MD23B)) :- t23b(V,MD23B),!.
```

```
t23b(1,md23b1).  
t23b(2,md23b2).  
t23b(3,md23b3).  
t23b(4,md23b4).
```

```
t23b(5,md23b5).
```

```
tabela(24,V-_,mess(X)):- t24(V,X),!.
```

```
t24(a,md24x).
```

```
t24(0,md24s).
```

```
tabela(26,V-_,(mess(md26x),write(X),mess(md26s),write(Y),nl,nl)) :-  
t26(V,X,Y).
```

```
t26(0,0,'NULA DU LOCAL').
```

```
t26(1,1,'MA''').
```

```
t26(2,2,'MA''').
```

```
t26(3,3,'MEDIA').
```

```
t26(4,4,'MEDIA').
```

```
t26(5,5,'DE BOA A 'OPTIMA').
```

```
tabela(D,X-R,true) :- ( X=0 ; mess(mimpos), write(D-X-R), nl,nl ), !.
```


MENSAGENS DO SISTEMA

MENSAGENS DO SISTEMA

Nao conheco pontos dentro dessa regio.

O equilibrio da vegetacao e' considerado

A sua pergunta incide sobre qual aptidao ?
Escreva um numero de 1 a 5 conforme o caso:

- 1-HABITACAO CONCENTRADA
- 2-AGRICULTURA INTENSIVA
- 3-AGRICULTURA NAO INTENSIVA
- 4-INDUSTRIA
- 5-INFRAESTRUTURAS DE RECREIO

Sobre cada um deles conheco:

- 1-a quantidade.
- 2-nomes das suas instanciaco es particulares.
- 3-do que dependem.
- 4-para o que contribuem.

Escreva os numeros correspondentes ao que lhe interessa ²conhecer,
separados por espaço.

Com que valor?

Os valores possiveis sao:

- MUITO APTO
- BASTANTE APTO
- APTO
- ALGO APTO
- POUCO APTO
- NAO APTO

Pretende explicacoes ? Responda sim ou nao.

A aptidao 7

selectivo 7

o descriptor 7

durante uns instantes asuande que eu me adapte a este ecran e que consulte a
se de conhecimentos

A media da aptidao

na regio limitada em
abscissas :

ordenadas :

tem o valor

foi calculado por

Deseja continuar a sessao?

Responda sim ou nao

adeus.

base de conhecimentos

ova ou

ntisa?

terminar com um ponto (.)

*** valor ilegal do descriptor

tem o valor 7

porque:

porque segundo o especialista:

com representatividade 7 -

Programa ORBI : especialista em ordenamento biofisico

Nucleo de Inteligencia Artificial

Departamento de Informatica

Universidade Nova de Lisboa

por substituição em

*** o factor 7

varia entre Plano e Modelado

iguais ou superiores a 28% e/ou a ocorrer um declive medio acentuado com coberto arboreo denso.

BAIXO

favoráveis à instalação humana definitiva

Os solos não são susceptíveis de qualquer utilização económica, apresentando severas a muito severas limitações para pastagens, exploração de matas e silvicultura. A aptidão pedológica à agricultura pode ser considerada muito baixa.

Os solos não são susceptíveis de utilização agrícola, salvo em casos especiais, apresentando poucas ou moderadas limitações para pastagens, exploração de matas e silvicultura. A aptidão pedológica à agricultura pode ser considerada baixa.

Os solos são susceptíveis de utilização agrícola de fraca intensidade. A aptidão pedológica à agricultura pode ser considerada média. Estes solos estão muito provavelmente, total ou parcialmente protegidos pelo artigo 1.º - n.º 1 do Dec. Lei 308/79.

Os solos são susceptíveis de utilização agrícola de moderada intensidade. A aptidão pedológica à agricultura pode ser considerada elevada. Estes solos estão, muito provavelmente, total ou parcialmente protegidos pelo artigo 1.º - n.º 2 do Dec. Lei 308/79.

Os solos são susceptíveis de utilização agrícola de grande intensidade. A aptidão pedológica à agricultura pode ser considerada muito elevada. Estes solos estão, muito provavelmente, total ou parcialmente protegidos pelo artigo 1.º - n.º 1 do Dec. Lei 308/79.

a Piro resistência vegetal é
MUITO ELEVADA

Por se tratarem de formações onde :

- o grau de humidade do solo no período crítico ;
- a biomassa combustível ;
- a humidade relativa no interior da formação ;
- e a combustibilidade directa das espécies que a constituem

determinam um grau de combustibilidade da formação MUITO BAIXO.

A potencialidade da paisagem para formas de recreio é
BAIXA

Por se tratar de uma unidade paisagística onde são compatíveis
14 de 57 actividades possíveis

A paisagem não se encontra degradada, ou a degradação existente não exige medidas especiais de recuperação.

[não foi fornecida ainda esta mensagem]

MUITO ELEVADA

Por se tratar de formações de calcários e/ou margas e/ou arenitos consolidados

Os aquíferos ocorrentes mais importantes apresentam um
ELEVADA

dado pertencerem ao manto Mio-Pliocénico cativo.

Não é conhecido.

MUITO ELEVADA

com regresso extremamente difícil ao do climax potencial

o subsolo não apresenta recursos minerais conhecidos ou os que apresenta não foram considerados.

a piro resistência vegetal e'

MEDIA

por se tratarem de formações onde :

- o grau de humidade do solo no período crítico ;
- a biomassa combustível ;
- a humidade relativa no interior da formação ;
- e a combustibilidade directa das espécies que a constituem

determinam um grau de combustibilidade da formação MEDIO

Muito detalhadas? (s/n)

Sempre que aparece o sinal barra (-) a iniciativa é sua para escrever uma frase. Esta pode ser uma pergunta ou uma afirmação.

Termine a frase com (.) ou (?) para uma resposta simples.

Termine com (+) para obter explicações não detalhadas.

Termine com (++) para obter explicações detalhadas (incluindo descritores

Termine com (*) para obter o registo da sessão na impressora.

Pode combinar (*) com (+) ou (++) .

Acabe sempre com (RETURN).

O risco potencial de incêndio corrido é'

ELEVADO

devido a ocorrerem temperaturas máximas no estio (corridas com o gradiente altimétrico e a exposição das encostas)

IGUAIS OU SUPERIORES a 28 graus C.

e/ou a ocorrer um declive médio acentuado com coberto arbóreo denso.

O risco potencial de incêndio corrido é'

BAIXO

devido a ocorrerem temperaturas máximas no estio (corridas com o gradiente altimétrico e a exposição das encostas)

INFERIOR a 27 graus C.

ou ocorrer um declive médio inferior a acentuado e/ou não existir coberto arbóreo denso.

O coberto vegetal ou é inexistente ou constitui fitocenoses de reduzido interesse florístico e/ou ecológico.

DE ROTURA DO EQUILIBRIO
com resso difícil ao climax potencial

Os solos são considerados como tendo riscos de erosão
MEDIOS

dado que o seu grau de erodibilidade e declive determinam limitações ao seu uso que em área representam valores inferiores a 50%.

O potencial em águas subterrâneas é
MUITO BAIXO

por se tratarem de recursos das formações impregiáveis do Paleogenico.

Especifique as coordenadas. [Termine com (.)]

Não definiu coordenadas!

Essas coordenadas não existem neste mapa!

É MODERADAMENTE ACENTUADO

ELEVADA

por se tratar de calcários onde um complexo de arenitos pouco considerados e areias

MEDIA

por se tratar de uma das seguintes situações :

- zona de declive suave a moderado, ou muito acentuado, exposta a poente
- zona de declive moderado/acentuado a acentuado, exposta a Nascente
- zona plana, em altura, abrigada dos ventos dominantes

Os aquíferos ocorrentes mais importantes apresentam uma protecção

MUITO BAIXA

contra a poluição dada a natureza fissurada dos terrenos que possibilita uma penetração directa dos agentes contaminadores sem filtração ou retenção.

A protecção dos aquíferos é considerada

MUITO ELEVADA

devido à não ocorrência de recursos em águas subterrâneas significativas.

A poluição afecta unicamente as águas superficiais.

Ocorrem fitocenoses de elevado interesse florístico muito pouco resistentes a acções externas que aconselham a sua protecção integral.

DE ROTURA DE EQUILIBRIO
com resso possível ao climax potencial

Os solos são considerados como tendo riscos de erosão

BAIXO

dado apresentarem como limitações as associadas à sua reduzida espessura efectiva.

O potencial em águas subterrâneas é

BAIXO

por se tratarem de recursos do gres do Cretácico com caudais unitários baixos

O declive médio

MEDIA

por se tratar de formações margosas e/ou de arenitos

pouco consolidados, puros ou em complexo

MUITO BAIXA

por se tratar de rochas duras impermeáveis, que não ocorrem nesta área de estudo

MUITO ELEVADA

por se tratar de formações dunares, coberturas arenosas ou aluviões

ELEVADA

por se tratar de uma das seguintes situações :

- zona de declive moderada/accentuado a accentuado, exposta a Poente
- zona de declive suave a moderado ou muito accentuado, exposta a Sul

Os aquíferos ocorrentes mais importantes apresentam uma protecção

BAIXA

contra a poluição dada a natureza arenosa dos terrenos cuja capacidade natural de filtração permite uma protecção parcial.

Não são conhecidos quaisquer valores de ordem geológica

Não existem aqui valores

considerados na definição das riquezas culturais da área de estudo.

PROXIMO

do climax potencial

Os solos são considerados como tendo risco de erosão

MUITO BAIXO

por não apresentarem quaisquer limitações ou cuja 'única limitação' é o excesso de 'água durante grande parte do ano.

O potencial em 'águas subterrâneas' é

MEDIO

por se tratarem de recursos dos calcários do Jurássico com caudais relativamente aleatórios.

O conjunto das características termo pluviométricas indicam condições

a resistência mecânica do substrato e

por se tratar de formações de calcários e/ou margas e/ou arenitos consolidados

ELEVADA

por se tratar de formações de arenitos consolidados ou complexos de calcários, margas e arenitos

E' ACENTUADO

BAIXA

por se tratar de formações margosas ou de arenitos consolidados, puros ou em complexo

MUITO BAIXA

por se tratar de uma zona de vale, exposta aos ventos dominantes

MUITO ELEVADA

por se tratar de uma zona de declive moderada/accentuado a moderado exposta a Sul

Os aquíferos ocorrentes mais importantes apresentam uma protecção

MEDIA
dado o seu reduzido potencial e variavel vulnerabilidade contra a poluicao.
Existem valores de ordem geologica complementares (Sistem

definicao das riquezas culturais da area de estudo.

SIMILAR
ao do clima potencial

O potencial em 'aguas subterraneas e'
ELEVADO
por se tratarem de recursos do aquifero cativo do Mio-Pliocenico mais
custosos relativamente 'a exploracao.

BAIXA
por se tratar de formacoes de dunas, coberturas arenosas
areias, arenitos pouco consolidados ou formacoes em complexo de
calcarios, margas gipsiferas e dolomias

MUITO ELEVADA
por se tratar de calcarios ou de complexos de calcarios e arenitos
consolidados

E' MUITO ACENTUADO

MEDIA
por se tratar de formacoes em complexo de calcarios, e/ou margas,
e/ou dolomitos (dolomias) e/ou arenitos

BAIXA
por se tratar de uma das seguintes situacoes :
- zona nao plana, exposta a norte e aos ventos dominantes
- zona de declive suave a moderado, exposta a nascente
- zona plana em altura nao abrisada dos ventos dominantes
- zona plana, em baixa, abrigada dos ventos dominantes

MEDIA
por se tratar de formacoes ou dunas, coberturas arenosas, areias, arenitos
pouco consolidados ou formacoes em complexo de calcarios, margas gipsiferas
e dolomias

Os aquiferos ocorrentes mais importantes apresentam uma proteccao
MEDIA
dado o seu reduzido potencial e variavel vulnerabilidade contra a poluicao.

ocorre nesta zona um elemento geologico considerado notav

pode inferir do equilibrio da vegetacao.

Os solos sao considerados como tendo riscos de erosao
MUITO ELEVADOS
dado que o seu grau de erodibilidade e declive determinam limitacoes
ao seu uso em 100% da 'area.

O potencial em 'aguas subterraneas e'
MUITO ELEVADO
por se tratar de recursos do aquifero cativo do Mio-Pliocenico facilmente
exploraveis.

o subsolo apresenta recursos em
AREIAS BRANCAS
com interesse.

A piro resistencia vegetal e'
MUITO BAIXA
por se tratar de formacoes onde :

- o grau de humidade do solo no periodo critico ;
- a biomassa combustivel ;
- a humidade relativa no interior da formacao ;
- e a combustibilidade directa das especies que a constituem

determinam um grau de combustibilidade da formacao MUITO ELEVADO

A piro resistencia vegetal e'
BAIXA
por se tratarem de formacoes onde :

- o grau de humidade do solo no periodo critico ;
- a biomassa combustivel ;
- a humidade relativa no interior da formacao ;
- e a combustibilidade directa das especies que a constituem

determinam um grau de combustibilidade da formacao ELEVADO.

A piro resistencia vegetal e'
ELEVADA
por se tratarem de formacoes onde :

- o grau de humidade do solo no periodo critico ;
- a biomassa combustivel ;
- a humidade relativa no interior da formacao ;
- e a combustibilidade directa das-especies que a constituem-----

determinam um grau de combustibilidade da formacao BAIXO.

A potencialidade da paisagem para formas de recreio e'
MUITO BAIXA
por se tratar de uma unidade paisagistica onde sao compativéis
12 de 57 actividades possíveis

A potencialidade da paisagem para formas de recreio e'
MEDIA
por se tratar de uma unidade paisagistica onde sao compativéis
23 a 27 de 57 actividades possíveis

A potencialidade da paisagem para formas de recreio e'
ELEVADA
por se tratar de uma unidade paisagistica onde sao compativéis

33 a 39 de 57 actividades Possiveis

A acessibilidade do territorio depende do grau de afastamento das principais vias de acesso existentes e da inclinacao media do terreno, tem o valor Nao e' considerado.

O escoamento pluvial superficial e'

Existem

(Arribas em altura media superior a 50 metros).

Jazida de fosseis do Miocenico (Praia do Penedo)

Jazida de pegadas de Dinossauros (

actual (mesmo que de reduzido ou nulo interesse floristico e' susceptivel de alterar o estado de equilibrio das fitocenoses que, com interesse floristico, ocorrem nas areas envolventes.

Ocorrem fitocenoses de caracteristicas antropozoosenas cuja alteracao do seu estado actual (mesmo que de reduzido ou nulo interesse floristico e' susceptivel de alterar o estado de equilibrio das fitocenoses que com interesse floristico ocorrem nas areas envolventes.

Ocorrem fitocenoses de caracteristicas antropozoosenas de elevado interesse floristico susceptiveis, no entanto, de serem utilizadas como pastagens e/ou culturas arvenses de sequeiro (montados)

Ocorrem fitocenoses de caracteristicas naturais de elevado interesse floristico susceptiveis, no entanto, de terem uma utilizacao moderada.

Ocorrem fitocenoses de caracteristicas antropozoosenas de elevado interesse floristico que aconselham um forte condicionamento das accoes susceptiveis de alterarem o seu uso equilibrado.

Ocorrem fitocenoses de caracteristicas naturais de elevado interesse floristico que aconselham um forte condicionamento das accoes susceptiveis de alterarem o seu uso equilibrado.

A potencialidade cinegetica e paisagistica dependente da sensibilidade visual da area e da probabilidade de ocorrencia de especies animais com interesse cinegetico e'

Nao sao conhecidos quaisquer valores de ordem arqueologica.

Existem elementos arqueologicos de interesse atribuveis

'a Idade do Bronze'

A potencialidade da paisagem para formas de recreio e'
MUITO ELEVADA

Por se tratar de uma unidade paisagistica onde sao compativeis
42 de 57 actividades possiveis

A paisagem encontra-se degradada por se tratar de uma das seguintes situacoes :

- exploracao de inertes a céu aberto sem plano de reconversao ;
- sistema dunal com sobre-utilizacao humana ;

- 'area semi-agricola fortemente erosionada' ~~existindo medidas especiais de recuperacao.~~

podendo ser considerada

'A Idade do Ferro'

~~'a Época Cartaginesa'~~

'a Época Romana'

'a Época Muculmana'

o Forte do Cavalo

o Castelo de Sesim

ELEVADOS

dado que o seu grau de erodibilidade e declive determinam limitacoes ao seu uso em 100% da 'area.

dado que as características da bacia hidrografica onde se encontra incluido e a pluviosidade media anual determinam um numero de aspereza corrido

DIGITALIZAÇÃO

.10.

```
10 REM .....
15 REM ...
20 REM ... PROGRAMA PARA DIGITALIZACAO DE DADOS TERRITORIAIS. ...
25 REM ...
30 REM .....
35 REM ...
40 REM ... VITOR LOPES DIAS ...
45 REM ... LUIS MONIZ PEREIRA ...
50 REM ...
55 REM ... DEPARTAMENTO DE INFORMATICA ...
60 REM ... UNIVERSIDADE NOVA DE LISBOA ...
65 REM ... QUINTA DA TORRE ...
70 REM ... 2825 MONTE DA CAPARICA ...
75 REM ...
80 REM .....
85 REM ...
90 REM ... - QUANDO DECLARAR UM NOVO FICHEIRO, ...
95 REM ... CERTIFIQUE-SE QUE E O ULTIMO. CASO ...
100 REM ... CONTRARIO DESTUIRA OS SEGUINTEs. ...
105 REM ...
110 REM .....
115 INIT
120 PAGE
125 X=62
130 Y=62
135 WINDOW 1,X+1,-2,Y+9
140 X3=(X-1)/4000
145 Y3=(Y-1)/4000
150 P=1
155 X9=X*Y
160 X8=X9+12+6
165 REM AS 12 POSICOES ^ SAO PARA GUARDAR AS COORDENADAS REAIS DA ORIGEM
170 REM AS 6 POSICOES SAO PARA O TITULO
175 DIM A$(X8),C$(1),F$(X8),P$(1),L$(7),T$(7),D$(1),E$(1),V$(1),I$(1)
180 C$=""
185 A$=""
190 FOR X4=1 TO X8-1
195 A$=A$&C$
200 NEXT X4
205 F$=A$
210 PRINT "QUAL O NUMERO DO FICHEIRO A UTILIZAR? (MAIOR QUE 3)"
215 INPUT F
220 IF F<4 THEN 210
225 PRINT "NOVO FICHEIRO? (S OU N)."
230 INPUT B$
235 FIND F
240 IF B$<>"S" THEN 385
245 MARK 1,X8.
250 PRINT "QUAL O TITULO DO MAPA? (MAX. 6 CARACTERES)"
255 INPUT T$
260 IF LEN(T$)>6 THEN 250
265 F$=REP(T$,X9+13,LEN(T$))
270 PRINT @33,0:0,0,1
275 FIND F
280 INPUT @33:Y$
285 Y$=REP(" ",22,12)
290 Y$=REP(T$,28,LEN(T$))
295 FIND F
300 PRINT @33:Y$
305 PRINT @33:"S"
310 PRINT @33,0:0,0,0
315 CLOSE
320 PRINT "QUAIS AS COORDENADAS DO CANTO INFERIOR ESQUERDO DO MAPA?"
325 PRINT "COORDENADA X? (MAX. 6 CARACTERES)"
330 INPUT L$
335 IF LEN(L$)>6 THEN 325
```

```

340 F$=REP(CL$,X9+1,LEN(CL$))
345 PRINT "COORDENADA Y? (MAX. 6 CARACTERES)"
350 INPUT L$
355 IF LEN(L$)>6 THEN 345
360 F$=REP(CL$,X9+7,LEN(CL$))
365 FIND F
370 PRINT @33:F$
375 CLOSE
380 GO TO 390
385 INPUT @33:F$
390 PRINT "QUAL O VALOR DOS DADOS?"
395 INPUT V$
400 PRINT "QUAL A REPRESENTATIVIDADE? (1 A 5)."
405 INPUT I1
410 IF I1<1 OR I1>5 THEN 400
415 IF "1">V$ OR V$>"5" THEN 430
420 I=VAL(V$)
425 GO TO 485
430 IF I1=5 THEN 445
435 PRINT "VALOR NAO COMPATIVEL COM A REPRESENTATIVIDADE"
440 GO TO 390
445 IF V$<>"0" THEN 460
450 I=6
455 GO TO 485
460 IF "A"<=V$ AND V$<="K" THEN 475
465 PRINT "VALOR ";V$;" NAO E PERMITIDO."
470 GO TO 390
475 I=ASC(V$)-58
480 REM ** NA INSTRUCAO SEGUINTE, 46 DETERMINA O CARACTER ASCII.
485 D=(I1-1)*5+I+46
490 HOME
495 PAGE
500 PRINT "OPCOES: 1-ALTERAR VALOR E REPR. ";
505 PRINT "2-MESA (AMARELO-MARCA, VERDE-APAGA, AZUL-PREENCHE, BRANCO-MENU";
510 PRINT ") 3-APAGAR ECRAN ";
515 PRINT "4-GUARDAR DADOS, S/ SOBREPOSICOES "
520 PRINT "5-SOBREPOR 6-VER ZONA TRABALHO ";
525 PRINT "7-APAGAR Z.TRAB. 8-VER FICHEIRO 9-DESENHA Z.TRABALHO ";
530 PRINT "10-DESENHAR FICHEIRO 11-OPCAO PLOTTER EM PARALELO 12-FIM"
535 INPUT I
540 GO TO I OF 390,570,1395,1255,1430,1415,1475,1040,1505,1525,1540,1560
545 HOME
550 PAGE
555 PRINT "CODIGO INCORRECTO. CORRIJA"
560 GO TO 500
565 REM *** INPUT DO CONTORNO ***
570 INPUT @8:X4,Y4,P$
575 IF P$="8" THEN 570
580 INPUT @8:X4,Y4,P$
585 IF P$="0" THEN 580
590 X4=INT(X4*X3+1.5)
595 Y4=INT(Y4*Y3+1.5)
600 IF P$="8" THEN 490
605 IF P$="4" THEN 700
610 IF Y4<1 THEN 580
615 IF X4<1 THEN 580
620 MOVE X4,Y4
625 T=X4+(Y4-1)*X
630 IF P$="2" THEN 665
635 IF P=1 THEN 650
640 MOVE @1:X4,Y4
645 PRINT @1:"+"
650 PRINT "+"
655 A$=REP(" ",T,1)
660 GO TO 580
665 A$=REP(" ",T,1)

```

.12.

```
670 IF P=1 THEN 685
675 MOVE @1:X4,Y4
680 PRINT @1:"0"
685 PRINT "0"
690 GO TO 580
695 REM *** IDENTIFICACAO E PREENCHIMENTO ***
700 Z=POSCAS,"1",X)
705 IF Z=0 THEN 770
710 Z1=POSCAS,"1",Z+1)
715 IF Z1=0 OR Z1>X9-X THEN 770
720 IF INT(Z/X)<>INT(Z1/X) THEN 760
725 IF Z1=Z+1 THEN 760
730 Y4=Z+1-X
735 X4=POSCAS," ",Y4)
740 IF X4<>0 AND X4<Z1-X-1 THEN 760
745 FOR B=Z+1 TO Z1-1
750 A$=REPC"2",B,1)
755 NEXT B
760 Z=Z1
765 GO TO 710
770 Z=X+1
775 Z=POSCAS,"2",Z+1)
780 IF Z=0 THEN 935
785 IF POSCAS," ",Z-1)=Z-1 OR POSCAS," ",Z+X)=Z+X THEN 800
790 IF POSCAS," ",Z+1)=Z+1 OR POSCAS," ",Z-X)=Z-X THEN 800
795 GO TO 775
800 X4=Z
805 X4=X4+X
810 IF X4>X9 THEN 835
815 D$=SEG(CAS$,X4,1)
820 IF D$="1" THEN 835
825 A$=REPC" ",X4,1)
830 GO TO 805
835 X4=Z
840 X4=X4-1
845 D$=SEG(CAS$,X4,1)
850 IF D$<>"1" THEN 840
855 Y4=POSCAS,"1",Z+1)
860 FOR B=X4+1 TO Y4-1
865 A$=REPC" ",B,1)
870 NEXT B
875 Y4=Z
880 X4=X4-X
885 Y4=Y4-X
890 IF Y4<=X THEN 915
895 D$=SEG(CAS$,Y4,1)
900 IF D$="1" THEN 915
905 A$=REPC" ",Y4,1)
910 GO TO 885
915 Z=Y4+X-2
920 IF X4=>Z THEN 775
925 Z=X4
930 GO TO 775
935 Z=0
940 Z=POSCAS,"2",Z+1)
945 IF Z=0 THEN 960
950 A$=REPC"1",Z,1)
955 GO TO 940
960 GOSUB 980
965 HOME
970 GO TO 500
975 REM *** VIZUALIZACAO DO VECTOR (MATIZ IMPLICITA) ***
980 PAGE
985 W=32
990 A=0
995 A=POSCAS,"1",A+1)
```

```

1000 IF A=0 THEN 1030
1005 Y4=INT(A/X)+1
1010 X4=A-(Y4-1)*X
1015 MOVE @W:X4,Y4
1020 PRINT @W:"+"
1025 GO TO 995
1030 RETURN
1035 REM *** VIZUALIZACAO DO CONTEUDO DO FICHEIRO EM TRATAMENTO ***
1040 W=32
1045 PAGE
1050 MOVE @W:1,Y+5
1055 PRINT @W:"CAR."
1060 MOVE @W:1,Y+3
1065 PRINT @W:"VALOR"
1070 MOVE @W:1,Y+1
1075 PRINT @W:"REPR."
1080 Z9=7
1085 FOR I=47 TO 83
1090 A=0
1095 Z=0
1100 C$=CHR(I)
1105 Z=POS(F$,C$,Z+1)
1110 IF Z=0 THEN 1150
1115 IF Z>X9 THEN 1150
1120 A=1
1125 Y4=INT(Z/X)+1
1130 X4=Z-(Y4-1)*X
1135 MOVE @W:X4,Y4
1140 PRINT @W:C$
1145 GO TO 1105
1150 IF A=0 THEN 1245
1155 MOVE @W:Z9,Y+5
1160 PRINT @W:C$
1165 I1=I-46
1170 IF I1<=25 THEN 1205
1175 Y4=5
1180 IF I1<>26 THEN 1195
1185 I$="0"
1190 GO TO 1215
1195 I$=CHR(I1+38)
1200 GO TO 1215
1205 Y4=INT((I1-1)/5)+1
1210 I$=CHR(I1-(Y4-1)*5+48)
1215 MOVE @W:Z9,Y+3
1220 PRINT @W:I$
1225 I$=CHR(Y4+48)
1230 MOVE @W:Z9,Y+1
1235 PRINT @W:I$
1240 Z9=Z9+1
1245 NEXT I
1250 GO TO 535
1255 REM *** STOCKAGEM DOS DADOS EM TRATAMENTO ***
1260 PAGE
1265 REM **IDENTIFICACAO DE SOBREPOSICOES **
1270 C$=" "
1275 P$=CHR(D)
1280 I2=0
1285 Z=0
1290 Z=POS(C$,"1",Z+1)
1295 IF Z=0 THEN 1340
1300 X4=POS(F$,C$,Z)
1305 IF X4<>Z THEN 1325
1310 F$=REPC(P$,Z,1)
1315 A$=REPC(" ",Z,1)
1320 GO TO 1290
1325 I2=1

```


.14.

```
1330 GO TO 1290
1335 REM ** STOCAGEM DOS ELEMENTOS SEM PROBLEMAS **
1340 FIND F
1345 PRINT @33:F$
1350 CLOSE
1355 IF I2=0 THEN 490
1360 REM *** TRATAMENTO DAS SOBREPOSICOES ***
1365 HOME
1370 PAGE
1375 PRINT "PONTOS COM CONFLITO. DECIDA OS QUE QUER SOBREPOSTOS."
1380 GOSUB 985
1385 GO TO 535
1390 REM *** APAGAR ECRAN ***
1395 HOME
1400 PAGE
1405 GO TO 535
1410 REM *** VIZUALIZAR ZONA DE TRABALHO ***
1415 GOSUB 985
1420 GO TO 535
1425 REM *** STOCKAGEM, S/VERIFICAR SOBREPOSICOES ***
1430 Z=0
1435 P$=CHR(D)
1440 I2=0
1445 Z=POS(A$, "1", Z+1)
1450 IF Z=0 THEN 1335
1455 F$=REP(P$, Z, 1)
1460 A$=REP(" ", Z, 1)
1465 GO TO 1445
1470 REM *** APAGAR ZONA DE TRABALHO ***
1475 C$=" "
1480 A$=" "
1485 FOR I=1 TO X9-1
1490 A$=A$&C$
1495 NEXT I
1500 GO TO 490
1505 REM *** DESENHAR ZONA DE TRABALHO ***
1510 W=1
1515 GOSUB 990
1520 GO TO 490
1525 REM *** DESENHAR FICHEIRO ***
1530 W=1
1535 GO TO 1045
1540 REM *** ALTERACAO DA OPCAO PLOTTER EM PARALELO ***
1545 P=P*-1
1550 GO TO 535
1555 REM *** FIM DO PROGRAMA ***
1560 CLOSE
1565 STOP
```

CONTROLADOR DO TRAÇADOR GRÁFICO

```

.mcell .gelfd, .drbes, .drast, .fork, .drfin, .drend
.iif ndf mms$t, mms$t=0
.iif ndf erl$s, eri$s=0
.iif ndf tim$it, tim$it=0
.iif ndf fl$vec, fl$vec=314
.iif ndf fl$csr, fl$csr=176514
.iif ndf fl$pri, fl$pri=4

        ioerr = 1
        flsts = 20000
        flsiz = 0
        fl$cod = 377
        .gelfd
        .drbes fl, fl$vec, flsiz, flsts

        mov     flcqe, r4
        bsl     a$wont(r4)
        bcc     flerr
        beg     fldun
flret:   bis     #100, @#fl$csr
        return

;int service routine
        .drast fl, fl$pri

        mov     flcqe, r4
        lst     @#fl$csr
        bmi     flret
        bic     #100, @#fl$csr
        .fork   flfork

flnxt:   tstb    @#fl$csr
        bpl     flret
.iif ea  mms$t
        movb   @a$buff(r4), @#fl$csr+2
        inc    a$buff(r4)
.iiff
        jsr    pc, @#stbyt
        mov    (sp)+, @#fl$csr+2
.endc

        inc    a$wont(r4)
        beg    fldun
        br     flnxt
flerr:   bis     #ioerr, @a$csw(r4)
fldun:   .drfin fl
flfork:  .word  0,0,0,0
        .drend fl
        .end

```

INTERFACES

```
/* GOG.PL */
```

```
sos :- repeat, trimcore, see('mo:F'), read(E), seen, E, fail.
```

```
imp :- assert(impfine).
```

```
saida(G) :- !P(G), fail.
```

```
saida(G) :- G.
```

```
!P(G) :- ( impfine, tell('!P:'),  
          ( mP1(C) ; solve(G) ), nl, told ; true ), !.
```

```
mP(mess(F)) :- see(F), repeat, setC(C), ( C=26, seen, ! ; put(C), fail ),
```

```
volta :- tell('mo:F'), write(fail), put(46), told, fail.
```

```
volta :- sos.
```

```
cont([adeus]) :- halt.
```

```
cont(LCM) :- sram(LCM), retractall(mens(L)), ! ; retract(mens(G)), G.
```

```
sram([E|LCM]) :-
```

```
( try( retract(acordo) ),
```

```
  try( retract(elipse(FQ,EXS,EL)) ),
```

```
  in(e,E), !,
```

```
  ( elipse(FQ,EL,QI,LCM,[]),
```

```
/* [frase eliptica] :
```

```
    extras_el(EXS,QI,Q) ;
```

```
    saida(mess(s2)), Q=fail ) ;
```

```
  ( trimcore,
```

```
    frase([E|LCM],Q,[]);
```

```
/* [outras frases] :
```

```
    saida(mess(s2)), Q=fail ) ), !,
```

```
tell('mo:F'), write(exec(Q)), put(46), told.
```

```
/* GOMA.PL */
```

```
/*TOP LEVEL*/
```

```
ola:- erase, reverse, mess(m50a), nonev, mess(m50b), fail.  
ola:- repeat, see(user), nl, soma.
```

```
soma :- trimcore, core, nl, so, fail.
```

```
soma :- see('m:B'), read(S), seen, S, !, fail.
```

```
/* GOMA.PL */
```

```
/*TOP LEVEL*/
```

```
ola:- erase, reverse, mess(m50a), norev, mess(m50b), fail.
```

```
ola:- repeat, see(user), nl, some.
```

```
some :- trimcore, core, nl, go, fail.
```

```
some :- see('a:B'), read(S), seen, S, !, fail.
```

```
/* ECRAN VT100 */
```

```
ecrfx:-erase,screev,write('resra para o calculo de '),nl,!.  
ecrmov:-reverse,write('quando poder apasar escreva s '),  
        ler([s]),scnovev,erase,!.  
csi:-put(27),put(91).  
erase:-csi,put(50),put(74),pcurs.  
raferase:-csi,put(49),put(74),pcurs.  
reverse:-csi,put(55),put(109).  
novev:-csi,put(48),put(109).  
marg:-csi,put(50),put(48),put(59),put(50),put(52),put(114).  
pcurs:-csi,put(48),put(59),put(48),put(72).  
npcurs:-csi,put(50),put(49),put(59),put(48),put(72),put(13).  
resetar:-csi,put(48),put(59),put(48),put(114).  
numcarM:-csi,put(63),put(51),put(104).  
numcarM:-csi,put(63),put(51),put(108).  
ecrfixo:-scnovev,reverse,pcurs,numcarM.  
ecrmov2:-novev,mars2,npcurs2.  
mars2:-csi,put(54),put(114).      /* resiao scroll 6 linhas abaixo */  
npcurs2:-csi,put(55),put(59),put(48),put(72).  
numlinM:-csi,put(63),put(57),put(104).  
numlinM:-csi,put(63),put(57),put(108).  
setab:-put(27),put(72).  
cltab:-csi,put(51),put(103).  
screev:-csi,put(63),put(53),put(104).  
scnovev:-csi,put(63),put(53),put(108).
```


?-of(247,xfy,seiao).
?-of(245,xfy,se).
?-of(243,xfy,ou).
?-of(241,xfy,e).
?-of(239,xfy,onde).
?-of(238,fx,com).
?-of(100,fx,?).
?-of(35,xfy,?).
?-of(30,xfy,-).

```
ii:-i_.  
ii:- retract((i_ :- _)), retractall(ii), seen, trimcore, sos.  
i_:-consult(sra),consult(sinn),consult(semn),consult(sos),!,fail.  
?-ii.
```

```
1_ :- consult(soma), consult(moa), consult(orbi), consult(dia), consult(dia)
      consult(baco), consult(files), consult(vt100), !, fail.
```

```
11 :- 1_.
```

```
11 :- retract((1_ :- _)), retractall(ii), seen, trimcore, ola.
```

```
?- ii.
```

FR SY:FLEXM.SAV
R FLXMM

FR SY:FLXMM.SAV
R FLEXM

FR SY:FLXM.SAV
R FLXM

PRI (sog,sinn,sra,semn,moa,din).pl
PRI (dia,soma,orbi,baco,metac,menu).pl
PRI (aquis,filins,palins,lins,analys,dilins).pl
PRI (selins,balins,vt100,ex1,ex2,ex3).pl
PRI (ex4,ex5,elipse,extra,oper,files).pl
PRI (smorbi,b,f).pl
PRI *.com

ARMAZENAMENTO DOS DADOS DIGITALIZADOS NA
BASE DE DADOS


```
program formarBD (finp,foutp);
```

```
(*Programa PASCAL para ler varios ficheiros contendo cada um os valores cor-  
respondentes a um mapa diferente da mesma regioao e formar uma base de dados  
de clausulas unarias PROLOG contendo cada uma delas as coordenadas de um pon-  
to e os valores correspondentes a esse ponto nos diferentes mapas *)
```

```
var nomeinp,nomeout:string;  
    finp,foutp:file of char;  
    m:packed array [1..800,1..23] of char;(*limites inferiores aos reais*)  
    c,t:char;  
    w1,l,f,z,v,r,x1,y1,x2,y2,fmax,lmax:integer;  
    fim:boolean;
```

```
procedure coordenadas;
```

```
    var i1,i2,j1,j2:integer;
```

```
    begin
```

```
        writeln('entrei na procedure coordenadas');
```

```
        if c=' ' then
```

```
            begin
```

```
                read(finp,i1,j1,i2,j2);
```

```
                write(i1:1,j1:1,i2:1,j2:1)
```

```
            end
```

```
        else
```

```
            begin
```

```
                writeln('faltou o espaco de separacao');
```

```
                i1:=ord(c)-ord('0');
```

```
                read(finp,c);
```

```
                while (c>='0') and (c<='9') do
```

```
                    begin
```

```
                        i1:=i1*10+ord(c)-ord('0');
```

```
                        read(finp,c)
```

```
                    end;
```

```
                read(finp,j1,i2,j2)
```

```
            end;
```

```
            if f=1 then
```

```
                begin
```

```
                    writeln('primeira file de input');
```

```
                    x1:=i1; x2:=i2; y1:=j1; y2:=j2
```

```
                end
```

```
            else
```

```
                if (i1<>x1) or (i2<>x2) or (j1<>y1) or (j2<>y2) then
```

```
                    begin
```

```
                        writeln('Esta file diz respeito a outros pontos');
```

```
                    halt
```

```
                    end
```

```
        end;(*fim do procedimento coordenadas*)
```

```

begin (* inicio do programa principal *)
writeln('nome da file de output ?');
read(nomeoutp);
rewrite(foutp,nomeoutp);
writeln('Quantos descriptors ,i.e. mapas, vao ser lidos ?');
readln(fmax);
for f:=1 to fmax do

    begin
    writeln('Nome da file de input ?');
    readln(nomeinp);
    reset(finp,nomeinp); (*abre file *)
    l:=1; fim:=false;
    repeat
        while not eoln(finp) do (* no caso de existirem <cr> *)

            begin
            read(finp,c);
            if(c >='/') and (c<='S') then
                begin
                m[l,f]:=c;
                write(c);
                l:=l+1
                end

            else
                begin
                coordenadas;
                writeln('sai de coordenadas');
                fim:=true
                end

            end;
            write('sai do while');
            if (not fim) then
                readln(finp)

        until fim;
        close(finp);
        writeln('fechou finp')
    end; (* fim do ciclo for *)

lmax:=l-1; (* lmax = numero de pontos lidos,i.e.linhas da matriz *)
writeln(lmax);
writeln(foutp,x1:1,y1:1,x2:1,y2:1);(*no inicio ficam as coordenadas dos pontos limites *)
w1:=x1;
for l:=1 to lmax do

    begin
    write(foutp,'p(',x1:1,',',y1:1,',(dp(');
    for f:=1 to fmax do

        begin (*esta descodificacao podia fazer-se com 'case of' *)
        ti='y';
        if ord(m[l,f])<ord('H') then

```

```

begin
z:=ord(m[l,f])-ord('/');
r:=z div 5+1;
v:=z mod 5+1
end

else

begin
if ord(m[l,f])=ord('H') then

begin
v:=0;
r:=5
end

else

begin
t:=chr(ord(m[l,f])+24);
r:=5
end

end;
if t='y' then
write(foutp,v:1,'-',r:1)
else
write(foutp,t:1,'-',r:1);
if f=14 then
write(foutp,')',ds(')
else

begin
if f=fmax then
write(foutp,')')
else
write(foutp,',')
end

end;

writeln(foutp,')'); (*escreveu-se uma clausula unaria*)
x1:=x1+2; (*incremento das coordenadas dependendo da propria digitali-
zacao *)
if x1>x2 then

begin
x1:=w1;
y1:=y1+2
end

end; (*escreveu-se a base de dados *)
write('escreveu a BD');
close(foutp,lock)
end. (*fim do programa principal *)

```

```

*****
/*
/*      Preparacao das sub-matrizes de pontos
/*
/******

pontos(Xm,XM,YM) :- base(Xb,Yb),
                    slot(Xm,Xb,X), slot(XM,Xb,Xk), slot(YM,Yb,Y), Y10 is Y-10,
                    see(pontos), read(F), asserta(next(F)),
                    construct(X,Xk,Y,Y10).

slot(X,Xb,N) :- Xr is X-Xb, N is Xr-(Xr mod 10)+1.

construct(,_,_,_,-) :- next(?-end), seen.

construct(X,Xk,Y,Y10) :- retract(next(F)), asserta(F), scan(Y),
                        build(X,Xk,Y,Y10).

construct(X,Xk,Y,-) :- Y_10 is Y-10, construct(X,Xk,Y_10,Y).

scan(Y) :- read(F), check_p(F,Y), !.

check_p(F,Y) :- last_p(F,Y), asserta(next(F)).

check_p(F,Y) :- assertz(F), scan(Y).

last_p(?-end,-).

last_p(o(_ ,Y,_),Ym) :- Y < Ym.

build(X,Xk,Y,Y10) :- X =< Xk, X10 is X+10, build(X,X10,Xk,Y,Y10).

build(X,X10,-,Y,Y10) :- locate(X,Y,F), show(F), save(X,X10,Y,Y10,F), fail.

build(-,X,Xk,Y,Y10) :- trimcore, core, build(X,Xk,Y,Y10).

save(X,X,-,-,-) :- told.

save(X,NX,Y,Y10,F) :- save(Y,Y10,X,F), X2 is X+2, save(X2,NX,Y,Y10,F), !.

save(Y,Y,-,-).

save(Y,NY,X,F) :- save(X,Y,F), Y2 is Y+2, save(Y2,NY,X,F).

save(X,Y,F) :- retract(o(X,Y,D)), tell(F), write(o(X,Y,D)), write(' '), nl.

save(-,-,-).

locate(X,Y,F) :- Xr is X/10, Yr is Y/10,
                 CX1 is Xr/10+48, CX2 is (Xr mod 10)+48,
                 CY1 is Yr/10+48, CY2 is (Yr mod 10)+48,
                 name(F,[114,CX1,CX2,CY1,CY2]).

```

```
show(F) :- tell(user), write(F), nl.
```

```
base(0,0).
```

BASE DE DADOS COM CERCA DE 600 PONTOS E SEUS
DESCRIPTORES DUMA REGIÃO DE SESIMBRA

- .r0870.
- .r0970.
- .r1070.
- .r0869.
- .r0969.
- .r1069.
- .r0868.
- .r0968.
- .r1068.
- .r0867.
- .r0967.
- .r1067.
- .r0866.
- .r0966.
- .r1066.
- .r0865.
- .r0965.
- .r1065.
- .r0864.
- .r0964.
- .r1064.
- .r0863.
- .r0963.
- .r1063.
- .r0862.
- .r0962.

~~Handwritten scribbles~~
~~Handwritten scribbles~~
.r0861.

d(81,871,d(d(1,3,3,5,5,5,2,5,3,5,5,5),d(5,5,a,2,2,5,0,5,2,3,5,5),d(4,5,2,5,1,5,4,3,4,5,4),d(4,5,0,5,1,4,0,5,0,5))).
d(81,873,d(d(1,4,3,5,5,5,2,5,3,5,5,5),d(5,5,0,5,2,5,0,5,3,4,5,5),d(4,5,2,5,1,5,4,3,4,5,4),d(4,5,0,5,1,5,0,5,0,5))).
d(81,875,d(d(1,2,3,5,5,5,2,5,3,5,3,4),d(1,1,0,5,2,5,0,5,3,5,5,5),d(4,5,2,5,1,5,5,5,5,5,5),d(4,5,0,5,1,5,0,5,0,5))).
d(81,877,d(d(3,3,1,3,5,5,2,5,3,5,3,4),d(5,5,0,5,2,5,0,5,2,3,5,5),d(4,5,2,5,1,5,5,5,5,5,5),d(4,5,0,5,1,5,0,5,0,5))).
d(81,879,d(d(4,3,1,4,5,5,2,5,3,5,3,3),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,5,1,5,5,5,5,5,5),d(4,5,0,5,1,5,0,5,0,5))).
d(83,871,d(d(2,5,3,5,5,5,2,4,3,5,5,5),d(5,5,a,5,2,3,0,5,3,5,5,5),d(4,5,2,3,1,4,4,3,4,a,2),d(4,4,0,5,1,2,0,5,2,1))).
d(83,873,d(d(1,4,3,5,5,4,2,4,3,5,5,4),d(5,4,a,4,2,4,0,5,3,5,5,5),d(4,5,2,3,1,4,4,3,4,a,2),d(4,5,0,5,1,4,0,5,0,5))).
d(83,875,d(d(3,4,3,5,5,5,2,5,3,5,5,4),d(1,1,a,3,2,5,0,5,3,5,5,5),d(4,5,2,5,1,5,5,4,4,5,5),d(4,5,0,5,1,5,0,5,0,5))).
d(83,877,d(d(3,4,3,5,5,5,2,5,3,5,5,4),d(5,5,0,5,2,5,0,5,2,2,5,5),d(4,5,2,5,1,5,5,4,5,5,5),d(4,5,0,5,1,5,0,5,0,4))).
d(83,879,d(d(3,4,3,5,5,5,2,5,3,5,5,5),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,5,1,5,5,4,4,5,5),d(4,5,0,5,1,5,0,5,0,4))).
d(85,871,d(d(1,3,3,5,2,5,3,4,5,4,5,5),d(5,5,a,5,0,5,0,5,3,5,5,5),d(4,5,3,5,3,4,2,2,2,c,2),d(4,5,0,5,4,5,0,5,2,4))).
d(85,873,d(d(1,2,3,5,5,3,2,3,3,3,5,5),d(5,5,a,5,0,5,0,5,3,5,5,5),d(4,5,4,4,2,4,2,4,2,c,2),d(4,5,0,5,4,5,0,5,0,4))).
d(85,875,d(d(2,5,3,5,5,5,2,3,3,5,5,5),d(5,5,a,5,2,3,0,5,3,5,5,5),d(4,5,4,3,1,4,2,4,c,5,2),d(4,5,0,5,1,3,0,5,1,3))).
d(85,877,d(d(2,5,3,5,5,5,2,5,3,5,5,5),d(5,5,a,3,2,5,0,5,2,3,5,5),d(4,5,2,5,1,4,4,3,4,a,4),d(4,5,0,5,1,4,0,5,1,5))).
d(85,879,d(d(1,2,3,5,5,5,2,5,3,5,5,5),d(5,5,0,4,2,5,0,5,1,5,5,5),d(4,5,2,5,1,4,4,3,4,a,4),d(4,5,0,5,1,5,0,5,1,5))).
d(87,871,d(d(1,3,3,5,2,4,3,4,5,4,5,5),d(5,5,a,5,0,5,0,5,3,5,5,4),d(4,4,3,4,3,4,1,0,4,c,1),d(4,5,0,5,4,5,0,5,2,5))).
d(87,873,d(d(2,3,3,5,5,4,2,4,3,4,5,5),d(5,5,a,5,0,5,0,5,3,4,5,5),d(4,5,4,5,2,4,2,4,2,c,2),d(4,5,0,5,4,5,0,5,2,3))).
d(87,875,d(d(2,5,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,3,5,5),d(4,5,4,5,2,4,2,4,1,c,2),d(4,5,0,5,4,5,0,5,1,2))).
d(87,877,d(d(2,5,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,2,2,0,5,1,5,5,5),d(4,5,4,4,2,4,4,4,2,c,3),d(4,5,0,5,4,5,0,5,1,4))).
d(87,879,d(d(2,4,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,2,3,0,5,1,5,5,5),d(4,5,4,4,2,4,5,3,2,a,3),d(4,5,0,5,4,5,0,5,1,4))).
d(89,871,d(d(2,5,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,2,2,4,5),d(2,4,4,5,3,4,1,0,4,c,1),d(4,5,0,5,4,5,0,5,2,5))).
d(89,873,d(d(2,5,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,4,4,5,2,0,5,0,5,0,5),d(4,5,0,5,4,5,0,5,2,4))).
d(89,875,d(d(2,5,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,5,0,5,0,5),d(4,5,0,5,4,5,0,5,1,4))).
d(89,877,d(d(1,3,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,3,1,2,1,c,2),d(4,5,0,5,4,5,0,5,1,2))).
d(89,879,d(d(1,3,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,4,5,2,4,c,4),d(4,5,0,5,4,5,0,5,3,4))).

o(101,671,d(d(3,4,4,5,4,5,2,5,3,5,5,3),d(5,5,a,4,0,5,0,5,3,5,5,4),d(4,4,1,4,1,4,3,4,1,c,2),d(2,4,0,5,4,5,0,5,4,2))).
o(101,673,d(d(4,2,4,5,4,4,2,4,3,5,5,3),d(5,5,a,4,0,5,0,5,1,3,5,4),d(4,4,1,4,1,4,3,1,1,c,3),d(2,4,0,5,4,4,0,5,1,4))).
o(101,675,d(d(2,2,4,5,5,3,2,4,3,5,5,3),d(5,5,a,5,0,5,0,5,1,5,5,3),d(2,3,3,3,1,3,3,1,4,c,2),d(2,4,0,5,4,4,0,5,1,2))).
o(101,677,d(d(2,4,4,5,5,4,2,5,3,5,3,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,4,2,3,1,3,4,1,3,0,5),d(2,5,0,5,4,5,0,5,5,4))).
o(101,679,d(d(1,4,4,5,4,4,2,4,3,5,5,4),d(5,5,a,4,0,5,0,5,1,5,5,5),d(4,5,1,4,1,3,2,1,2,a,1),d(2,3,0,5,2,4,0,5,5,5))).
o(103,671,d(d(4,3,4,5,4,5,2,5,3,5,3,3),d(5,5,a,4,0,5,0,5,3,4,4,4),d(2,4,1,5,1,4,2,4,2,4,2,c,1),d(2,5,0,5,4,5,0,5,3,4))).
o(103,673,d(d(2,4,4,5,4,4,2,5,3,5,3,3),d(5,5,a,5,0,5,0,5,1,5,4,5),d(2,5,1,4,1,3,3,1,4,c,2),d(2,5,0,5,4,5,0,5,1,3))).
o(103,675,d(d(2,5,4,5,5,4,2,4,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,4,5),d(2,5,3,5,1,3,2,1,3,c,1),d(2,5,0,5,4,5,0,5,4,5))).
o(103,677,d(d(2,5,4,5,5,4,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,1,5,4,3),d(2,4,3,3,1,3,2,1,3,0,5),d(2,5,0,5,4,4,0,5,5,2))).
o(103,679,d(d(1,2,4,5,4,4,2,4,3,5,5,3),d(5,5,a,4,0,5,0,5,1,5,5,5),d(4,5,4,3,5,0,5,0,5,a,1),d(4,2,0,5,2,5,0,5,5,1))).
o(105,671,d(d(1,3,4,5,5,4,2,4,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,3,3,1,3,3,1,2,0,5),d(2,5,0,5,4,5,0,5,3,4))).
o(105,673,d(d(1,4,4,5,5,4,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,3,3,4,2,1,2,1,2,0,5),d(2,5,0,5,4,5,0,5,5,3))).
o(105,675,d(d(1,4,4,5,5,5,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,3,5,1,0,4,0,4,0,5),d(2,5,0,5,4,5,0,5,4,5))).
o(105,677,d(d(2,3,4,5,5,4,2,5,3,5,2,2),d(5,5,a,2,0,5,0,5,3,4,5,4),d(4,4,1,4,1,0,4,0,4,a,1),d(2,3,0,5,4,3,0,5,4,3))).
o(105,679,d(d(3,3,4,5,5,5,2,3,3,5,5,3),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,1,4,5,0,5,0,5,a,1),d(4,5,0,5,2,3,0,5,0,5))).
o(107,671,d(d(1,3,4,4,5,5,2,4,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,3,5,2,4,3,4,3,c,3),d(2,5,0,5,4,5,0,5,3,5))).
o(107,673,d(d(2,3,4,3,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,3,5,2,3,2,1,2,c,1),d(2,5,0,5,4,5,0,5,3,3))).
o(107,675,d(d(1,3,4,3,5,5,2,5,3,5,5,4),d(5,5,a,3,0,5,0,5,3,5,4,4),d(2,4,3,4,2,0,4,0,5,a,1),d(4,3,0,5,4,5,0,5,4,3))).
o(107,677,d(d(1,3,4,4,5,5,2,4,3,5,3,3),d(5,5,a,2,0,5,0,5,3,4,5,3),d(2,3,1,5,1,0,5,0,5,a,1),d(4,5,0,5,4,4,0,5,4,2))).
o(107,679,d(d(3,5,4,1,5,5,0,4,0,4,0,4),d(0,4,0,4,0,5,0,5,1,5,0,4),d(0,4,0,4,0,5,0,5,0,5),d(0,4,0,5,0,4,0,5,4,1))).
o(109,671,d(d(2,3,4,5,5,5,2,5,3,5,5,4),d(5,5,a,4,0,5,0,5,3,5,4,5),d(2,5,3,4,2,4,5,4,5,c,5),d(2,3,0,5,4,5,0,5,3,5))).
o(109,673,d(d(1,4,4,5,5,5,2,5,3,5,3,3),d(5,5,0,4,0,5,0,5,3,5,4,4),d(2,3,1,4,1,0,4,4,2,a,2),d(4,3,0,5,4,5,0,5,3,2))).
o(109,675,d(d(4,3,4,3,5,5,2,3,3,5,2,2),d(5,5,a,3,0,5,0,5,3,3,5,3),d(4,2,1,3,1,0,5,0,5,a,1),d(4,3,0,5,4,4,0,5,0,4))).
o(109,677,d(d(4,5,4,2,5,5,0,4,0,5,0,4),d(5,2,0,4,0,5,0,5,0,4,0,4),d(0,4,0,4,0,5,0,4,0,5,0,5),d(0,4,0,5,0,4,0,5,5,1))).
o(109,679,d(d(2,2,4,3,5,4,2,3,3,3,5,3),d(0,4,a,5,0,5,0,5,1,3,0,4),d(2,3,4,3,2,3,3,3,2,0,5),d(3,2,0,5,0,3,0,5,5,3))).

a(81,661,d(d(5,3,3,5,2,5,4,5,5,5,3,3),d(5,5,0,5,0,5,0,5,4,5,5,4),d(4,3,1,4,2,4,2,2,1,c,2),d(3,5,0,5,4,5,0,5,1,2))).
a(81,663,d(d(2,4,3,5,4,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,5,5),d(4,5,4,5,2,4,3,2,2,c,2),d(3,5,0,5,4,5,0,5,1,4))).
a(81,665,d(d(2,3,3,5,4,4,2,4,3,5,5,4),d(5,5,a,4,0,5,0,5,3,5,5,5),d(4,5,4,5,2,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,1,3))).
a(81,667,d(d(3,3,3,5,5,5,2,4,3,5,2,3),d(5,5,a,3,0,5,0,5,3,5,5,5),d(4,5,4,5,4,1,3,1,c,1),d(3,5,0,5,4,5,0,5,0,5))).
a(81,669,d(d(3,5,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,2,2,0,5,3,5,5,5),d(4,5,4,4,2,3,4,2,3,a,1),d(4,3,0,5,4,4,0,5,1,3))).
a(83,661,d(d(5,2,3,5,2,3,4,4,5,4,3,3),d(5,5,a,2,0,5,0,5,4,3,5,3),d(2,3,4,3,2,3,2,2,0,5),d(3,5,0,5,4,5,0,5,2,3))).
a(83,663,d(d(2,3,3,5,4,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,5,4),d(4,3,4,5,2,4,4,2,5,c,4),d(3,5,0,5,4,5,0,5,2,5))).
a(83,665,d(d(1,3,3,5,5,4,1,3,3,5,5,5),d(5,5,a,3,0,5,0,5,3,5,5,5),d(4,5,4,5,4,4,1,0,4,c,1),d(3,5,0,5,4,5,0,5,2,4))).
a(83,667,d(d(3,4,3,5,5,5,2,5,3,5,5,4),d(5,5,a,4,0,5,0,5,3,5,5,5),d(4,5,4,5,2,4,3,3,3,c,3),d(3,5,0,5,4,5,0,5,2,4))).
a(83,669,d(d(3,3,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,5,5),d(4,5,4,5,2,4,4,3,4,c,4),d(3,5,0,5,4,5,0,5,2,4))).
a(85,661,d(d(2,3,3,5,4,5,2,4,3,4,5,4),d(5,5,a,4,0,5,0,5,3,4,4,5),d(2,5,4,5,2,3,4,2,4,0,5),d(3,5,0,5,4,5,0,5,2,3))).
a(85,663,d(d(1,5,3,5,4,5,2,5,3,5,5,5),d(5,5,a,4,0,5,0,5,3,5,4,3),d(2,4,4,5,2,4,2,2,4,c,2),d(3,5,0,5,4,5,0,5,2,5))).
a(85,665,d(d(3,3,3,5,5,4,1,3,3,5,5,5),d(5,5,a,3,0,5,0,5,3,5,5,4),d(4,3,4,5,4,4,3,2,3,c,3),d(3,5,0,5,4,5,0,5,2,5))).
a(85,667,d(d(2,4,3,5,5,5,2,5,3,5,5,4),d(5,5,0,5,0,5,0,5,3,5,4,4),d(2,4,4,5,2,4,3,3,2,c,4),d(3,4,0,5,4,5,0,5,2,5))).
a(85,669,d(d(1,3,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,5,5),d(4,4,4,4,2,4,4,2,3,c,4),d(4,4,0,5,4,5,0,5,2,5))).
a(87,661,d(d(1,4,3,5,4,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,5,2,3,5,2,5,0,5),d(3,5,0,5,4,5,0,5,3,3))).
a(87,663,d(d(1,4,3,5,5,3,2,4,3,5,5,5),d(5,5,a,4,0,5,0,5,3,5,4,5),d(2,5,4,5,4,3,2,2,3,0,4),d(3,5,0,5,4,5,0,5,4,4))).
a(87,665,d(d(3,3,3,5,5,5,2,3,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,5,4,4,3,2,4,c,3),d(3,4,0,5,4,5,0,5,4,4))).
a(87,667,d(d(2,5,3,5,5,5,2,5,3,5,5,5),d(5,5,0,5,0,5,0,5,3,5,4,5),d(2,5,4,5,2,4,3,1,2,c,4),d(4,4,0,5,4,5,0,5,4,4))).
a(87,669,d(d(2,3,3,5,5,4,2,4,3,4,5,5),d(5,5,a,5,0,5,0,5,3,5,5,4),d(2,4,4,4,3,1,1,2,1,c,1),d(4,5,0,5,4,5,0,5,2,5))).
a(89,661,d(d(2,4,3,5,4,4,2,4,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,5,4,3,2,2,4,0,5),d(4,3,0,5,4,5,0,5,3,5))).
a(89,663,d(d(2,4,3,5,5,4,2,4,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,5,4,4,2,2,3,c,2),d(4,5,0,5,4,5,0,5,3,4))).
a(89,665,d(d(2,4,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,5,2,4,4,2,4,c,4),d(4,5,0,5,4,5,0,5,4,5))).
a(89,667,d(d(2,5,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,5,2,2,4,4,3,0,5),d(4,5,0,5,4,5,0,5,4,5))).
a(89,669,d(d(2,3,3,3,5,5,2,3,3,5,5,4),d(5,3,a,5,0,5,0,5,3,2,4,5),d(2,3,4,5,2,2,1,4,2,0,5),d(4,3,0,5,4,3,0,5,2,4))).

00101,661,d(d(1,5,4,5,2,5,4,5,5,5,5,5),d(5,5,a,1,0,5,0,5,3,5,4,4),d(2,4,3,5,0,0,4,0,5,0,5),d(4,5,0,5,4,5,0,5,3,3))).

00101,663,d(d(3,2,4,5,4,4,2,4,3,4,5,4),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,2,4,2,3,3,2,3,0,5),d(4,5,0,5,4,5,0,5,3,5))).

00101,665,d(d(2,3,4,5,4,5,2,5,3,5,3,4),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,2,4,1,3,4,2,4,0,5),d(4,4,0,5,4,5,0,5,3,4))).

00101,667,d(d(1,3,4,5,4,5,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,2,4,1,2,4,4,3,0,5),d(4,4,0,5,4,5,0,5,3,4))).

00101,669,d(d(3,3,4,5,4,5,2,5,3,5,5,3),d(5,5,a,5,0,5,0,5,3,5,5,3),d(2,4,1,3,1,2,3,4,3,0,5),d(2,3,0,5,4,5,0,5,5,4))).

00103,661,d(d(2,3,4,5,4,4,2,4,3,4,3,3),d(5,5,a,1,0,5,0,5,3,5,4,4),d(2,3,1,4,1,2,2,3,1,0,5),d(4,5,0,5,4,5,0,5,3,2))).

00103,663,d(d(3,3,4,5,4,4,2,3,3,3,5,3),d(5,5,a,3,0,5,0,5,3,3,4,5),d(2,5,1,3,1,2,4,4,4,0,5),d(4,5,0,5,4,5,0,5,3,2))).

00103,665,d(d(4,3,4,4,4,5,2,4,3,4,3,2),d(5,4,a,3,0,5,0,5,3,3,4,4),d(2,3,1,5,1,2,4,4,4,0,5),d(2,3,0,5,4,4,0,5,0,4))).

00103,667,d(d(4,3,4,5,4,4,2,5,3,5,5,3),d(5,5,a,4,0,5,0,5,3,5,4,5),d(2,5,1,4,1,2,5,4,5,0,5),d(2,5,0,5,4,5,0,5,0,5))).

00103,669,d(d(4,3,4,5,4,4,2,4,3,5,3,3),d(5,5,a,4,0,5,0,5,3,5,4,4),d(2,4,1,4,1,2,4,4,4,0,5),d(2,5,0,5,4,5,0,5,3,1))).

00105,661,d(d(2,3,4,5,4,4,4,2,5,4,3,5),d(5,5,a,2,0,5,0,5,3,5,1,3),d(5,2,1,5,1,2,1,3,2,0,5),d(4,5,0,5,4,5,0,5,0,5))).

00105,663,d(d(1,3,4,4,4,5,2,3,3,4,2,3),d(5,4,a,3,0,5,0,5,3,4,4,3),d(5,2,1,5,1,2,0,5,0,5,0,5),d(4,4,0,5,4,4,0,5,0,4))).

00105,665,d(d(3,2,4,4,4,3,5,2,5,3,5,3),d(5,4,a,4,0,5,0,5,3,4,4,3),d(2,3,1,4,1,2,0,4,0,4,0,5),d(4,3,0,5,4,4,0,5,3,4))).

00105,667,d(d(2,4,4,4,5,4,1,4,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,3,1,2,2,4,2,0,5),d(2,3,0,5,4,5,0,5,3,5))).

00105,669,d(d(2,5,4,5,4,3,2,4,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,2,1,0,4,0,4,0,5),d(2,5,0,5,4,5,0,5,3,5))).

00107,661,d(d(1,3,4,5,4,4,5,3,5,5,3,4),d(5,5,0,5,0,5,0,5,3,5,1,3),d(5,3,1,5,1,2,2,3,2,0,5),d(4,5,0,5,4,5,0,5,1,3))).

00107,663,d(d(3,2,4,4,4,5,5,4,5,4,5,4),d(5,4,0,5,0,5,0,5,3,4,1,4),d(5,5,1,5,1,2,0,5,0,5,0,5),d(4,4,0,5,4,5,0,5,3,3))).

00107,665,d(d(1,3,4,4,4,3,5,3,5,3,5,4),d(5,4,0,5,0,5,0,5,3,4,1,3),d(5,3,1,3,1,2,0,5,0,5,0,5),d(4,4,0,5,4,4,0,5,3,5))).

00107,667,d(d(1,4,4,5,5,5,2,5,3,5,5,5),d(5,5,a,4,0,5,0,5,3,5,4,5),d(2,5,3,4,1,4,5,4,4,0,5),d(4,3,0,5,4,5,0,5,3,5))).

00107,669,d(d(1,4,4,5,5,4,2,3,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,3,3,1,4,5,4,5,4,5),d(2,5,0,5,4,5,0,5,3,5))).

00109,661,d(d(1,4,4,4,2,5,4,4,5,5,2,3),d(5,5,0,4,0,5,0,5,3,5,2,4),d(3,4,1,5,1,2,4,2,2,0,5),d(2,4,0,5,4,5,0,5,1,2))).

00109,663,d(d(1,3,4,3,2,3,5,3,5,3,5,3),d(5,5,0,4,0,5,0,5,3,5,1,4),d(5,4,1,5,1,2,1,0,4,0,5),d(4,4,0,5,4,5,0,5,1,4))).

00109,665,d(d(2,3,4,3,5,3,5,2,3,3,5,4),d(5,5,a,3,0,5,0,5,3,5,4,3),d(2,3,3,3,1,2,2,0,4,0,4),d(4,5,0,5,4,5,0,5,3,4))).

00109,667,d(d(2,3,4,4,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,3,5,1,4,5,4,4,0,5),d(4,3,0,5,4,5,0,5,3,5))).

00109,669,d(d(2,3,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,3,5,1,4,4,5,4,5),d(2,4,0,5,4,5,0,5,3,5))).

d(81,851,d(d(1,4,3,5,2,5,4,5,5,5,5,2),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,3,3,2,1,2,0,5),d(3,5,0,5,4,5,0,5,0,5))).
d(81,853,d(d(3,4,3,5,2,5,4,5,5,5,5,5),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,5,3,2,1,0,4,0,5),d(3,5,0,5,4,5,0,5,1,1))).
d(81,855,d(d(3,3,3,5,2,5,3,4,5,5,5,5),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,5,4,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,1,2))).
d(81,857,d(d(3,3,3,5,2,5,3,5,5,5,5,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,5,4,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,1,4))).
d(81,859,d(d(3,2,3,5,2,5,3,4,5,5,5,4),d(5,5,0,5,0,5,0,5,4,5,2,4),d(3,5,2,4,4,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,1,2))).
d(83,851,d(d(4,2,3,5,2,5,4,5,5,5,2,2),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,4,1,2,5,2,4,0,5),d(3,5,0,5,4,5,0,5,0,5))).
d(83,853,d(d(3,4,3,5,5,3,4,3,5,3,5,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,4,3,3,2,2,2,0,5),d(3,5,0,5,4,5,0,5,1,4))).
d(83,855,d(d(3,5,3,5,5,3,2,3,3,3,5,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,4,4,3,3,3,2,2,0,5),d(3,5,0,5,4,5,0,5,1,5))).
d(83,857,d(d(1,3,3,5,2,4,3,4,5,4,5,5),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,4,4,0,5,0,4,0,5),d(3,5,0,5,4,5,0,5,1,4))).
d(83,859,d(d(2,3,3,5,2,5,3,4,5,5,5,3),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,3,4,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,0,5))).
d(85,851,d(d(3,3,3,5,2,4,4,4,5,4,5,3),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,3,1,2,4,1,2,0,5),d(3,5,0,5,4,5,0,5,0,4))).
d(85,853,d(d(3,3,3,5,5,5,2,5,3,4,5,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,4,4,2,3,2,2,4,0,5),d(3,5,0,5,4,5,0,5,1,4))).
d(85,855,d(d(1,3,3,5,5,4,4,5,3,4,5,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,4,3,2,3,3,2,3,0,5),d(3,5,0,5,4,5,0,5,1,5))).
d(85,857,d(d(2,3,3,5,2,4,4,4,5,4,5,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,4,2,0,4,0,4,0,5),d(3,5,0,5,4,5,0,5,1,5))).
d(85,859,d(d(5,2,3,5,2,5,4,5,5,5,2,3),d(5,5,0,5,0,5,0,5,4,5,2,3),d(3,4,1,4,2,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,0,5))).
d(87,851,d(d(2,4,3,5,2,5,4,4,5,4,3,2),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,1,0,4,1,1,0,5),d(3,5,0,5,4,5,0,5,3,3))).
d(87,853,d(d(2,3,3,5,2,5,4,4,5,4,3,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,1,2,1,1,1,0,5),d(3,5,0,5,4,5,0,5,1,4))).
d(87,855,d(d(2,4,3,5,2,5,4,5,5,5,5,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,1,1,2,1,1,0,5),d(3,5,0,5,4,5,0,5,1,5))).
d(87,857,d(d(2,3,3,5,2,5,4,5,5,5,5,3),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,2,1,4,1,4,0,5),d(3,5,0,5,4,5,0,5,1,3))).
d(87,859,d(d(4,2,3,5,2,4,4,4,5,4,5,3),d(5,5,0,5,1,0,5,0,5,4,5,4,3),d(3,3,1,4,2,0,2,2,1,0,5),d(3,5,0,5,4,5,0,5,0,4))).
d(89,851,d(d(2,3,3,5,2,5,3,4,5,5,3,3),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,3,2,3,3,1,0,5),d(3,5,0,5,4,5,0,5,3,2))).
d(89,853,d(d(2,4,3,5,2,5,4,4,5,5,5,3),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,2,2,3,2,3,0,5),d(3,5,0,5,4,5,0,5,1,2))).
d(89,855,d(d(2,5,3,5,2,5,4,5,5,5,3,2),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,2,1,4,1,4,0,5),d(3,5,0,5,4,5,0,5,3,2))).
d(89,857,d(d(3,4,3,5,2,5,4,5,5,5,5,5),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,3,2,1,2,1,4,0,5),d(3,5,0,5,4,5,0,5,3,5))).
d(89,859,d(d(3,2,3,5,4,3,2,3,3,4,5,3),d(5,5,0,5,2,0,5,0,5,4,3,4,3),d(2,3,4,4,2,3,2,2,1,0,5),d(3,5,0,5,4,5,0,5,3,2))).

s(101,651,d(d(2,3,4,5,2,3,3,4,5,5,3,3),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,1,3,4,2,3,4,3,0,5),d(4,5,0,5,4,5,0,5,5,4))).
s(101,653,d(d(4,2,4,5,3,3,4,5,5,5,2,3),d(5,5,0,5,0,5,0,5,3,5,3,4),d(1,4,1,4,2,2,3,2,4,0,5),d(4,5,0,5,4,5,0,5,0,4))).
s(101,655,d(d(2,2,4,5,4,3,5,3,5,5,5,3),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,3,1,2,2,2,2,0,5),d(4,5,0,5,4,5,0,5,0,5))).
s(101,657,d(d(2,4,4,5,2,5,4,5,5,5,3,5),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,1,0,5,0,5,0,5),d(4,5,0,5,4,5,0,5,1,4))).
s(101,659,d(d(2,4,4,5,2,5,4,5,5,5,3,3),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,1,3,1,2,1,0,5),d(4,5,0,5,4,5,0,5,1,5))).
s(103,651,d(d(2,3,4,5,3,4,4,4,5,5,3,3),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,1,4,2,2,4,4,3,0,5),d(4,5,0,5,4,5,0,5,5,5))).
s(103,653,d(d(2,2,4,5,3,4,4,5,5,5,2,3),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,1,5,1,2,4,2,5,0,5),d(4,5,0,5,4,5,0,5,5,3))).
s(103,655,d(d(3,2,4,5,2,3,4,4,5,5,5,3),d(5,5,0,5,0,5,0,5,3,5,2,4),d(1,3,1,3,1,1,3,2,2,0,5),d(4,5,0,5,4,5,0,5,0,5))).
s(103,657,d(d(3,3,4,5,2,4,4,4,5,5,2,2),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,1,0,4,2,2,0,5),d(4,5,0,5,4,5,0,5,0,4))).
s(103,659,d(d(3,4,4,5,2,5,4,5,5,5,3,3),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,1,2,2,3,2,0,5),d(4,5,0,5,4,5,0,5,1,4))).
s(105,651,d(d(4,3,4,5,2,3,3,4,5,5,3,4),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,2,3,1,2,2,4,2,0,5),d(4,5,0,5,4,5,0,5,5,5))).
s(105,653,d(d(3,3,4,5,3,5,4,5,5,5,2,3),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,1,5,1,2,5,2,5,0,5),d(4,5,0,5,4,5,0,5,5,2))).
s(105,655,d(d(4,3,4,5,3,3,4,5,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,3,3),d(1,3,1,4,1,2,3,2,3,0,5),d(4,5,0,5,4,5,0,5,0,5))).
s(105,657,d(d(2,5,4,5,2,3,5,3,5,5,2,2),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,1,2,3,2,3,0,5),d(4,5,0,5,4,5,0,5,0,5))).
s(105,659,d(d(2,4,4,5,2,5,4,5,5,5,2,3),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,1,2,2,3,2,0,5),d(4,5,0,5,4,5,0,5,0,5))).
s(107,651,d(d(3,2,4,5,2,4,3,5,5,5,3,3),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,2,4,4,2,1,0,4,0,5),d(4,5,0,5,4,5,0,5,5,5))).
s(107,653,d(d(4,3,4,5,3,5,4,4,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,1,5,3,2,5,2,5,0,5),d(4,5,0,5,4,5,0,5,5,3))).
s(107,655,d(d(4,2,4,5,3,3,4,5,5,5,5,3),d(5,5,0,5,0,5,0,5,3,5,3,4),d(1,4,1,4,1,2,4,2,4,0,5),d(4,5,0,5,4,5,0,5,0,5))).
s(107,657,d(d(2,4,4,5,4,4,5,4,5,5,5,3),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,4,1,2,4,2,4,0,5),d(4,5,0,5,4,5,0,5,0,5))).
s(107,659,d(d(4,2,4,5,2,5,4,4,5,5,2,4),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,1,2,3,3,2,0,5),d(4,5,0,5,4,5,0,5,0,5))).
s(109,651,d(d(2,5,4,4,2,5,3,5,5,5,3,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,2,4,4,0,5,0,5,0,5),d(4,5,0,5,4,5,0,5,5,5))).
s(109,653,d(d(3,2,4,5,3,4,4,4,5,5,3,4),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,1,5,3,2,4,2,4,0,5),d(4,4,0,5,4,5,0,5,5,3))).
s(109,655,d(d(3,3,4,5,3,4,4,5,5,5,5,2),d(5,5,0,5,0,5,0,5,3,5,3,4),d(1,5,1,4,1,2,5,2,4,0,5),d(4,4,0,5,4,5,0,5,3,2))).
s(109,657,d(d(2,3,4,4,4,4,5,5,5,5,2,3),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,4,1,2,5,2,5,0,5),d(4,3,0,5,4,5,0,5,0,5))).
s(109,659,d(d(3,3,4,4,2,5,4,4,5,5,2,4),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,1,2,2,3,2,0,5),d(4,3,0,5,4,5,0,5,0,5))).

d(81,641,d(d(2,4,3,5,2,4,4,4,5,5,3,5),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,5,3,2,5,0,4,0,5),d(5,3,0,5,4,5,0,5,1,3))))).
d(81,643,d(d(3,3,3,5,2,3,4,5,5,5,3,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,3,3,2,5,2,1,0,5),d(3,4,0,5,4,5,0,5,1,3))))).
d(81,645,d(d(4,2,3,5,2,5,4,5,5,5,3,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,5,3,2,4,0,4,0,5),d(5,5,0,5,4,5,0,5,0,5))))).
d(81,647,d(d(3,3,3,5,2,5,4,5,5,5,5,3),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,4,1,1,3,1,4,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(81,649,d(d(1,3,3,5,2,5,4,5,5,5,5,3),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,1,1,4,1,4,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(83,641,d(d(4,3,3,5,4,3,5,4,5,5,5,2),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,3,3,2,3,2,2,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(83,643,d(d(3,3,3,5,2,4,4,4,5,5,3,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,4,3,2,3,2,2,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(83,645,d(d(3,3,3,5,2,5,4,5,5,5,3,2),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,1,1,4,1,4,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(83,647,d(d(4,3,3,5,2,5,4,5,5,5,3,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,1,1,5,1,5,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(83,649,d(d(4,3,3,5,2,5,4,5,5,5,5,3),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,1,1,4,1,4,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(85,641,d(d(2,4,3,5,3,4,4,4,5,5,2,2),d(5,5,0,5,0,5,0,5,4,5,2,3),d(1,3,1,5,2,2,1,2,4,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(85,643,d(d(3,3,3,5,4,4,5,3,5,5,2,2),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,3,2,2,2,2,2,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(85,645,d(d(4,3,3,5,2,3,4,3,5,5,3,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,3,3,2,4,2,4,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(85,647,d(d(4,4,3,5,2,5,4,5,5,5,3,5),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,3,2,4,2,4,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(85,649,d(d(4,4,3,5,2,5,4,5,5,5,2,2),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,1,2,5,2,4,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(87,641,d(d(2,3,3,5,2,3,3,3,5,5,5,3),d(5,5,0,5,0,5,0,5,4,5,3,5),d(1,5,3,3,4,2,3,2,3,0,5),d(3,5,0,5,4,5,0,5,5,2))))).
d(87,643,d(d(2,4,3,5,3,3,4,4,5,5,2,3),d(5,5,0,5,0,5,0,5,4,5,3,3),d(1,4,1,5,2,2,3,2,3,0,5),d(3,5,0,5,4,5,0,5,0,4))))).
d(87,645,d(d(4,5,3,5,4,3,5,3,5,5,3,3),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,4,1,3,1,1,2,1,3,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(87,647,d(d(4,5,3,5,2,3,3,3,5,5,3,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,3,3,2,4,2,4,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(87,649,d(d(4,3,3,5,2,5,3,4,5,5,3,2),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,3,2,5,2,3,0,5),d(3,5,0,5,4,5,0,5,0,5))))).
d(89,641,d(d(2,4,3,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,4,5,3,5),d(1,5,3,5,4,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5))))).
d(89,643,d(d(2,5,3,5,2,4,3,4,5,5,3,3),d(5,5,0,5,0,5,0,5,4,5,3,5),d(1,5,3,4,4,2,2,2,2,0,5),d(3,5,0,5,4,5,0,5,5,3))))).
d(89,645,d(d(4,3,3,5,3,4,4,5,5,5,3,4),d(5,5,0,5,0,5,0,5,4,5,3,4),d(1,4,1,5,2,2,4,2,4,0,5),d(3,5,0,5,4,5,0,5,0,4))))).
d(89,647,d(d(4,3,3,5,2,3,4,3,5,5,3,3),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,4,1,4,2,2,3,1,2,0,5),d(3,5,0,5,4,5,0,5,3,1))))).
d(89,649,d(d(2,4,3,5,2,3,5,3,5,5,3,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,3,2,2,4,3,2,0,5),d(3,5,0,5,4,5,0,5,3,2))))).

d(101,641,d(d(2,3,4,5,4,5,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,4,5,
2,5,2,5,f,5),d(3,5,0,5,4,4,0,5,5,3))).
d(101,643,d(d(3,3,4,5,4,4,5,5,5,5,2,3),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,4,4,
2,4,2,4,f,4),d(3,5,0,5,4,5,0,5,0,4))).
d(101,645,d(d(5,4,4,5,2,5,3,5,5,5,3,3),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,
2,1,0,4,a,2),d(3,5,0,5,4,5,0,5,5,5))).
d(101,647,d(d(2,4,4,5,2,5,3,5,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,
0,5,0,5,0,5),d(3,4,0,5,4,1,0,5,5,5))).
d(101,649,d(d(2,4,4,5,2,4,3,4,5,5,5,3),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,4,
2,1,2,1,0,5),d(4,5,0,5,4,5,0,5,5,5))).
d(103,641,d(d(1,3,4,5,4,5,5,5,5,5,5),d(5,5,0,5,0,5,a,3,3,5,3,5),d(1,5,4,5,
2,5,2,5,f,5),d(3,4,0,5,3,4,0,5,4,3))).
d(103,643,d(d(4,3,4,5,4,5,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,4,5,
2,5,2,5,f,5),d(3,5,0,5,4,5,0,5,5,2))).
d(103,645,d(d(5,2,4,5,2,3,5,3,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,3,
2,3,2,5,f,2),d(3,5,0,5,4,5,0,5,5,5))).
d(103,647,d(d(2,4,4,5,2,5,3,5,5,5,5,3),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,
0,5,0,5,0,5),d(3,4,0,5,4,5,0,5,5,5))).
d(103,649,d(d(2,5,4,5,2,5,3,5,5,5,5,3),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,4,
0,5,0,5,0,5),d(4,5,0,5,4,5,0,5,5,5))).
d(105,641,d(d(2,3,4,5,4,5,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,4,5,
2,4,2,5,f,5),d(5,4,0,5,3,5,0,5,4,5))).
d(105,643,d(d(2,3,4,5,4,5,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,4,5,
2,5,2,5,f,5),d(3,5,0,5,4,4,0,5,4,3))).
d(105,645,d(d(2,3,4,5,4,3,5,4,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,3,
2,3,2,4,f,4),d(3,5,0,5,4,5,0,5,5,5))).
d(105,647,d(d(2,3,4,5,2,5,3,5,5,5,5,3),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,
0,5,0,5,0,5),d(3,4,0,5,4,5,0,5,5,5))).
d(105,649,d(d(2,4,4,5,2,5,3,5,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,
0,5,0,5,0,5),d(4,5,0,5,4,5,0,5,5,5))).
d(107,641,d(d(3,2,4,5,4,5,5,5,5,5,5,4),d(5,5,0,5,0,5,a,2,3,5,3,5),d(1,5,4,5,
2,3,2,5,f,4),d(5,5,0,5,3,5,0,5,4,5))).
d(107,643,d(d(3,4,4,5,4,5,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,4,5,
2,5,2,5,f,5),d(3,5,0,5,4,4,0,5,4,3))).
d(107,645,d(d(3,3,4,5,4,4,5,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,4,4,
2,4,2,5,f,5),d(3,5,0,5,4,5,0,5,5,5))).
d(107,647,d(d(2,4,4,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,
0,5,0,5,0,5),d(3,4,0,5,4,5,0,5,5,5))).
d(107,649,d(d(2,5,4,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,
0,5,0,5,0,5),d(4,5,0,5,4,5,0,5,5,5))).
d(109,641,d(d(1,5,3,3,4,5,5,5,5,5,5,5),d(5,5,0,5,0,5,a,3,3,5,3,5),d(1,5,4,5,
2,4,2,5,f,4),d(5,5,0,5,3,5,0,5,4,5))).
d(109,643,d(d(1,5,4,5,4,5,5,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,4,5,
2,5,2,5,f,5),d(3,5,0,5,4,4,0,5,5,4))).
d(109,645,d(d(1,3,4,5,4,4,5,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,4,1,
2,4,2,5,f,5),d(3,5,0,5,4,5,0,5,5,5))).
d(109,647,d(d(2,5,4,5,2,5,3,4,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,
0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
d(109,649,d(d(2,5,4,5,2,3,3,4,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,2,3,
0,5,0,5,0,5),d(3,3,0,5,4,5,0,5,5,5))).

o(81,631,d(d(2,3,3,2,2,5,3,3,5,3,0,3),d(5,4,0,5,0,5,0,5,4,2,3,3),d(1,3,3,3,4,0,5,0,5,0,5),d(2,3,0,5,4,3,0,5,5,5))).
o(81,633,d(d(2,3,3,3,2,4,3,4,5,4,5,4),d(5,3,0,5,0,5,0,5,4,3,3,4),d(1,3,3,3,4,0,2,1,2,1,0,5),d(3,1,0,5,4,3,0,5,5,5))).
o(81,635,d(d(2,3,3,5,3,4,4,4,5,5,2,3),d(5,5,0,5,0,5,0,5,4,5,3,3),d(1,4,1,4,2,2,4,2,4,0,5),d(3,5,0,5,4,5,0,5,5,2))).
o(81,637,d(d(2,3,3,5,2,3,4,4,5,5,5,3),d(5,5,0,5,0,5,0,5,4,5,2,4),d(3,4,1,4,1,4,2,4,2,2,0,5),d(5,3,0,5,4,5,0,5,5,3))).
o(81,639,d(d(4,2,3,5,4,3,5,3,5,5,3,2),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,4,3,2,2,4,2,4,0,5),d(5,4,0,5,4,5,0,5,5,1))).
o(83,631,d(d(2,4,3,5,2,5,3,4,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,4,5,0,5,0,5,0,5),d(3,4,0,5,4,5,0,5,5,5))).
o(83,633,d(d(2,4,3,4,2,5,3,4,5,4,5,4),d(5,4,0,5,0,5,0,5,4,2,3,4),d(1,4,3,4,4,0,0,5,0,5,0,5),d(3,3,0,5,4,4,0,5,5,5))).
o(83,635,d(d(2,5,3,5,2,4,3,4,5,5,5,4),d(5,5,0,5,0,5,0,5,4,5,3,5),d(1,5,3,5,4,2,2,2,1,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(83,637,d(d(2,5,3,5,3,3,4,4,5,5,2,2),d(5,5,0,5,0,5,0,5,4,5,3,4),d(1,4,1,3,2,2,3,2,4,0,5),d(3,5,0,5,4,5,0,5,5,4))).
o(83,639,d(d(2,4,3,5,3,2,4,4,5,5,3,3),d(5,5,0,5,0,5,0,5,4,5,2,4),d(3,5,1,5,2,2,2,3,2,2,0,5),d(3,5,0,5,4,5,0,5,5,2))).
o(85,631,d(d(1,4,3,5,2,4,3,3,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,4,4,2,2,2,2,a,1),d(3,5,0,5,4,5,0,5,5,5))).
o(85,633,d(d(2,4,3,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,4,3,5),d(1,5,3,5,4,0,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(85,635,d(d(2,4,3,4,2,5,3,4,5,4,5,4),d(5,4,0,5,0,5,0,5,3,2,3,4),d(1,4,3,5,4,0,0,5,0,5,0,5),d(3,4,0,5,4,4,0,5,5,5))).
o(85,637,d(d(2,4,3,4,2,5,3,4,5,4,5,3),d(5,3,0,5,0,5,0,5,4,4,3,4),d(1,4,3,4,4,0,0,5,0,5,0,5),d(3,4,0,5,4,4,0,5,5,5))).
o(85,639,d(d(2,4,3,5,3,3,4,3,5,5,5,4),d(5,5,0,5,0,5,0,5,4,5,3,5),d(1,3,1,3,2,2,2,3,2,3,0,5),d(3,5,0,5,4,5,0,5,5,3))).
o(87,631,d(d(2,2,3,3,4,4,5,3,0,3,0,3),d(5,3,0,5,0,5,0,5,3,2,3,3),d(1,3,4,4,1,2,2,2,2,3,f,3),d(3,2,0,5,4,3,0,5,5,3))).
o(87,633,d(d(1,3,3,3,2,5,3,4,5,4,5,3),d(5,3,0,5,0,5,0,5,3,3,3,4),d(1,3,3,3,4,0,2,1,2,1,a,1),d(3,3,0,5,4,3,0,5,5,5))).
o(87,635,d(d(2,5,3,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,4,0,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(87,637,d(d(2,4,3,4,2,5,3,4,5,4,5,4),d(5,3,0,5,0,5,0,5,3,3,3,3),d(1,4,3,4,4,0,0,5,0,5,0,5),d(3,4,0,5,4,3,0,5,5,5))).
o(87,639,d(d(2,3,3,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,4,5,3,5),d(1,5,3,5,4,0,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,4))).
o(89,631,d(d(0,4,3,2,0,4,0,4,0,4,0,4),d(0,4,0,5,f,2,0,5,0,4,0,4),d(0,4,0,4,0,0,0,4,0,4,0,4),d(0,4,0,5,0,4,0,5,5,4))).
o(89,633,d(d(0,4,3,1,0,4,0,4,0,5,0,4),d(0,5,0,5,0,5,0,5,0,4,0,4),d(0,4,0,4,0,0,0,4,0,5,0,4),d(0,4,0,5,0,5,0,5,5,5))).
o(89,635,d(d(1,3,3,5,2,4,3,4,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,4,4,4,3,2,2,1,a,1),d(3,5,0,5,4,5,0,5,5,5))).
o(89,637,d(d(4,4,3,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,4,0,0,5,0,5,0,4),d(3,5,0,5,4,5,0,5,5,5))).
o(89,639,d(d(2,3,3,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,4,4,3,5),d(1,5,3,5,4,0,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).

a(81,621,d(d(2,3,3,5,4,5,5,5,5,5,1,3),d(5,5,0,5,3,2,0,5,5,5,3,5),d(1,5,4,5,1,3,3,3,3,a,3),d(5,4,0,5,2,3,0,5,0,5))).
a(81,623,d(d(2,5,3,5,4,5,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,4,3,5),d(1,5,4,5,1,3,3,2,5,e,4),d(5,5,0,5,4,5,0,5,0,5))).
a(81,625,d(d(2,5,3,5,4,5,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,4,5,1,3,3,2,5,e,3),d(3,5,0,5,4,5,0,5,0,5))).
a(81,627,d(d(2,3,3,5,3,4,5,5,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,2,3,3,2,2,2,4,f,3),d(3,5,0,5,4,5,0,5,5,3))).
a(81,629,d(d(4,3,3,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(0,4,3,4,4,5,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
a(83,621,d(d(2,3,3,5,4,5,5,5,5,5,1,4),d(5,5,0,5,3,4,0,5,5,5,3,5),d(1,5,4,5,1,4,4,3,4,a,4),d(4,4,0,5,2,4,0,5,0,5))).
a(83,623,d(d(2,4,3,4,4,5,5,4,5,5,5,4),d(5,4,0,5,3,1,0,5,5,3,3,5),d(1,4,4,5,1,3,3,2,4,e,3),d(5,4,0,5,4,4,0,5,0,5))).
a(83,625,d(d(2,5,3,5,4,5,5,5,5,5,5,5),d(5,5,0,5,0,5,0,5,4,1,3,5),d(1,5,4,5,1,3,2,2,5,f,3),d(5,3,0,5,4,5,0,5,0,5))).
a(83,627,d(d(2,4,3,5,4,4,5,4,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,4,4,1,3,4,2,5,f,4),d(3,5,0,5,4,5,0,5,3,1))).
a(83,629,d(d(1,3,3,5,2,3,5,4,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(0,4,3,3,3,3,2,2,2,a,1),d(3,5,0,5,4,5,0,5,5,5))).
a(85,621,d(d(2,5,3,5,4,5,5,5,5,5,1,5),d(5,5,0,5,3,5,0,5,5,5,3,5),d(1,5,4,5,1,3,5,3,5,5,3),d(4,5,0,5,2,5,0,5,0,5))).
a(85,623,d(d(2,4,3,4,4,5,5,4,5,5,1,4),d(5,4,0,5,3,3,0,5,5,4,3,5),d(1,4,4,5,1,3,4,3,4,a,3),d(4,4,0,5,2,4,0,5,0,5))).
a(85,625,d(d(2,5,3,5,4,5,5,5,5,5,5,4),d(5,5,0,5,f,1,0,5,5,4,3,5),d(1,5,4,5,1,3,4,2,4,e,5),d(5,4,0,5,4,5,0,5,3,3))).
a(85,627,d(d(2,5,3,5,4,5,5,5,5,5,5,5),d(5,5,0,5,f,1,0,5,4,3,3,5),d(1,5,4,5,1,3,3,2,5,e,3),d(5,4,0,5,4,5,0,5,3,5))).
a(85,629,d(d(2,3,3,5,4,4,5,4,5,5,5,5),d(5,5,0,5,0,5,0,5,4,2,3,5),d(0,4,4,4,1,2,3,2,4,f,4),d(3,4,0,5,4,5,0,5,3,3))).
a(87,621,d(d(2,5,3,5,4,5,5,5,5,5,1,5),d(5,5,0,5,3,5,0,5,5,5,3,5),d(1,5,4,5,1,4,5,3,5,5,5),d(4,5,0,5,2,5,0,5,0,5))).
a(87,623,d(d(2,5,3,5,4,5,5,5,5,5,1,5),d(5,5,0,5,3,5,0,5,5,5,3,5),d(1,5,4,5,1,4,5,3,5,5,3),d(4,5,0,5,2,5,0,5,0,5))).
a(87,625,d(d(2,4,3,5,4,5,5,5,5,5,1,4),d(5,5,0,5,3,4,0,5,5,5,3,5),d(1,5,4,5,1,3,4,3,4,a,4),d(4,4,0,5,2,4,0,5,0,5))).
a(87,627,d(d(2,4,3,5,4,5,5,5,5,5,3,3),d(5,4,0,5,f,5,0,5,5,5,3,5),d(1,5,4,5,1,3,4,2,4,e,4),d(5,4,0,5,4,4,0,5,3,2))).
a(87,629,d(d(2,5,3,5,4,5,5,5,5,5,5,5),d(5,4,0,5,f,3,0,5,5,5,3,5),d(0,4,4,5,1,3,3,2,4,e,4),d(5,5,0,5,4,5,0,5,3,5))).
a(89,625,d(d(2,5,3,4,4,5,5,5,5,5,1,5),d(5,5,0,5,3,5,0,5,5,5,3,5),d(1,5,4,5,1,3,5,3,5,5,3),d(4,5,0,5,2,5,0,5,0,5))).
a(89,627,d(d(2,4,3,4,4,5,5,5,5,5,1,5),d(5,5,0,5,f,2,0,5,5,5,3,5),d(1,5,4,5,1,4,3,3,4,a,4),d(4,4,0,5,2,5,0,5,0,5))).
a(89,629,d(d(2,5,3,5,4,5,5,5,5,5,1,4),d(5,5,0,5,f,5,0,5,5,5,3,5),d(0,4,4,5,1,3,4,2,4,e,4),d(5,3,0,5,2,4,0,5,3,2))).

o(91,627,d(d(2,4,4,5,4,5,5,5,5,5,1,5),d(5,5,0,5,3,5,a,1,5,5,3,5),d(1,5,4,3,1,4,5,3,5,5,4),d(4,5,0,5,2,5,0,5,0,5))).
o(91,629,d(d(2,5,4,3,4,5,5,5,5,5,1,5),d(5,5,0,5,f,3,0,5,5,5,3,5),d(0,4,2,4,1,4,4,3,4,a,4),d(4,5,0,5,2,5,0,5,0,5))).
o(93,627,d(d(3,3,4,5,4,5,5,5,5,5,1,5),d(5,5,0,5,3,5,a,3,5,5,3,5),d(1,5,4,4,1,4,5,3,5,5,4),d(4,5,0,5,2,5,0,5,0,5))).
o(93,629,d(d(2,3,4,5,4,5,5,5,5,5,1,2),d(5,5,0,5,f,1,a,3,5,5,3,5),d(1,5,2,4,1,4,5,3,5,a,4),d(4,5,0,5,2,5,0,5,0,5))).
o(95,629,d(d(4,3,4,5,4,5,5,5,5,5,1,5),d(5,5,0,5,3,5,0,5,5,5,3,5),d(1,5,4,2,1,4,5,3,5,5,4),d(4,5,0,5,2,5,0,5,0,5))).

d(81,619,d(d(4,4,3,5,4,5,5,5,5,1,5),d(5,5,0,5,3,5,0,5,5,5,3,5),d(1,5,4,5,1,5,5,3,5,5,3),d(4,5,0,5,2,5,0,5,0,5))).
d(83,619,d(d(4,5,3,5,4,5,5,5,5,1,5),d(5,5,0,5,3,5,0,5,5,5,3,5),d(1,5,4,5,1,5,5,3,5,5,5),d(4,5,0,5,2,5,0,5,0,5))).

d(83,701,d(d(3,5,1,5,0,5,0,5,0,5,5,5),d(5,5,0,5,0,5,0,5,1,5,5,5),d(4,5,0,5,0,5,0,5,0,5,0,5),d(3,5,0,5,3,5,a,3,0,5))),
d(83,703,d(d(3,4,1,5,0,5,0,5,0,5,5,5),d(5,5,0,5,0,5,0,5,1,5,5,5),d(4,5,0,5,0,5,0,5,4,3,0,5),d(3,5,0,5,3,5,a,1,0,5))),
d(83,705,d(d(4,5,1,5,0,5,0,5,0,5,3,5),d(5,5,0,5,0,5,0,5,1,5,5,5),d(4,5,0,5,0,5,0,5,0,5,0,5),d(3,5,0,5,3,5,0,5,0,5))),
d(83,707,d(d(4,5,1,5,0,5,0,5,0,5,3,5),d(5,5,0,5,0,5,0,5,1,5,5,5),d(4,5,0,5,0,5,0,5,0,5,0,5),d(3,5,0,5,3,5,0,5,0,5))),
d(83,709,d(d(4,5,1,5,0,5,0,5,0,5,2,3),d(5,5,0,5,0,5,0,5,1,5,5,5),d(4,5,0,5,0,5,0,5,0,5,0,5),d(3,5,0,5,3,5,0,5,0,5))),
d(85,701,d(d(3,2,1,4,0,3,0,3,0,3,2,2),d(5,5,0,4,2,3,0,5,1,5,5,5),d(4,5,2,1,1,4,2,5,2,a,2),d(5,3,0,5,3,4,a,5,0,5))),
d(85,703,d(d(3,5,1,5,0,3,0,3,0,3,5,4),d(1,1,0,5,2,3,0,5,1,5,5,5),d(4,5,2,4,1,4,2,0,5,a,3),d(4,3,0,5,3,4,a,4,0,5))),
d(85,705,d(d(4,3,1,3,5,4,2,4,3,4,3,3),d(5,5,0,5,2,4,0,5,1,5,5,5),d(4,5,2,4,1,4,2,5,2,a,4),d(4,3,0,5,1,4,a,3,0,5))),
d(85,707,d(d(3,4,1,4,5,4,2,5,3,4,3,2),d(5,5,0,5,2,3,0,5,1,5,5,5),d(4,5,2,5,1,5,3,5,2,5,4),d(4,4,0,5,1,4,a,1,0,5))),
d(85,709,d(d(4,3,1,5,5,4,2,5,3,4,5,3),d(5,5,0,5,2,3,0,5,1,5,5,5),d(4,5,2,4,1,5,3,5,2,5,3),d(4,4,0,5,1,4,0,5,0,5))),
d(87,701,d(d(1,5,1,3,4,3,2,4,3,4,5,5),d(5,5,a,3,2,1,0,5,1,5,5,5),d(4,5,5,3,5,5,2,5,2,5,2),d(5,3,0,5,1,2,a,1,1,2))),
d(87,703,d(d(4,4,1,3,5,4,2,3,3,4,5,5),d(1,2,a,3,2,2,0,5,1,5,5,5),d(4,5,2,3,2,4,2,5,2,a,2),d(4,4,0,5,3,2,a,1,0,4))),
d(87,705,d(d(3,2,4,5,5,5,2,4,3,5,5,4),d(1,3,0,5,2,4,0,5,1,5,5,5),d(4,5,2,5,1,4,4,3,3,a,4),d(4,5,0,5,1,4,0,5,0,5))),
d(87,707,d(d(2,5,4,5,5,5,2,5,3,5,5,4),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,5,1,4,4,3,3,a,4),d(4,5,0,5,1,5,0,5,0,5))),
d(87,709,d(d(2,5,1,3,5,5,2,5,3,5,5,5),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,5,1,5,3,5,3,5,2),d(4,5,0,5,1,5,0,5,0,5))),
d(89,701,d(d(1,2,4,5,5,3,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,2,2,1,3,1,4,0,5),d(3,5,0,5,2,2,0,5,1,5))),
d(89,703,d(d(4,2,4,5,5,3,2,3,3,3,5,4),d(1,1,a,4,0,5,0,5,1,5,5,5),d(4,5,4,3,5,1,3,5,2,0,5),d(3,4,0,5,2,3,0,5,1,5))),
d(89,705,d(d(4,3,4,5,5,5,2,4,3,5,5,4),d(1,4,0,5,2,2,0,5,1,5,5,5),d(4,5,2,4,1,5,4,3,3,5,4),d(4,5,0,5,1,3,0,5,1,2))),
d(89,707,d(d(2,5,4,5,5,5,2,5,3,5,5,5),d(1,1,0,5,2,5,0,5,1,5,5,5),d(4,5,2,5,1,5,5,4,4,4,5,4),d(4,5,0,5,1,5,0,5,0,4))),
d(89,709,d(d(2,5,4,5,5,5,2,5,3,5,5,5),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,5,1,5,4,3,4,3,5,2),d(4,5,0,5,1,5,0,5,1,4))).

d(91,701,d(d(2,3,4,5,5,3,2,5,3,5,5,3),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,5,3,5,3,3,1,2,4,0,5),d(3,5,0,5,2,2,0,5,1,5))).
d(91,703,d(d(5,2,4,5,2,3,3,3,5,4,3,4),d(1,2,a,5,0,5,0,5,1,5,5,5),d(4,5,2,4,2,4,1,3,1,3,0,5),d(3,4,0,5,2,3,0,5,1,5))).
d(91,705,d(d(4,3,4,5,5,5,2,5,3,5,5,5),d(1,4,a,2,2,1,0,5,1,5,5,5),d(4,5,4,4,2,4,5,2,3,2,5,2),d(4,4,0,5,2,4,0,5,1,4))).
d(91,707,d(d(2,5,4,5,5,5,2,5,3,5,5,5),d(1,1,0,5,2,5,0,5,1,5,5,5),d(4,5,2,3,2,5,4,3,4,5,4),d(4,3,0,5,1,5,0,5,1,5))).
d(91,709,d(d(2,5,4,4,5,5,2,5,3,5,5,5),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,3,2,5,4,3,3,5,a,3),d(3,4,0,5,1,5,0,5,1,5))).
d(93,701,d(d(3,2,4,5,5,3,3,3,3,3,5,5),d(5,5,a,5,0,5,0,5,1,4,5,5),d(4,5,2,4,5,0,4,0,4,0,5),d(3,5,0,5,4,4,0,5,1,5))).
d(93,703,d(d(2,3,4,5,2,3,3,3,5,4,3,3),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,2,4,5,0,4,1,1,2,0,5),d(3,4,0,5,4,4,0,5,1,5))).
d(93,705,d(d(2,5,4,5,5,5,2,5,3,5,5,5),d(5,5,a,4,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,0,5,1,3,0,5),d(3,3,0,5,2,5,0,5,1,5))).
d(93,707,d(d(2,5,4,4,5,5,2,5,3,5,5,5),d(5,5,a,4,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,0,4,0,4,0,5),d(2,4,0,5,4,5,0,5,1,5))).
d(93,709,d(d(2,4,2,4,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,1,3,1,3,a,1),d(2,5,0,5,4,5,0,5,1,5))).
d(95,701,d(d(2,3,4,5,2,4,3,4,5,4,5,5),d(1,2,a,5,0,5,0,5,2,3,5,5),d(4,5,2,4,2,5,1,1,0,4,0,5),d(3,3,0,5,4,5,0,5,1,4))).
d(95,703,d(d(3,4,4,5,5,4,2,5,3,4,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,1,1,1,2,0,5),d(2,5,0,5,4,5,0,5,1,5))).
d(95,705,d(d(2,4,4,4,5,5,2,4,3,4,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,0,5,0,4,0,5),d(2,5,0,5,2,5,0,5,1,4))).
d(95,707,d(d(2,3,2,4,5,5,2,4,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,0,5,0,5,0,5),d(2,5,0,5,4,5,0,5,3,3))).
d(95,709,d(d(3,5,2,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,1,3,1,2,a,1),d(2,5,0,5,4,5,0,5,3,4))).
d(97,701,d(d(3,2,4,5,5,5,1,3,3,5,5,5),d(1,2,a,5,0,5,0,5,2,4,5,5),d(4,5,4,5,3,0,3,3,2,3,0,5),d(3,3,0,5,4,5,0,5,1,4))).
d(97,703,d(d(3,5,4,3,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,3,3,1,2,0,5),d(2,5,0,5,4,5,0,5,1,4))).
d(97,705,d(d(3,5,2,5,5,3,2,4,3,4,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,0,5,0,5,0,5),d(2,5,0,5,2,5,0,5,3,4))).
d(97,707,d(d(3,4,2,5,5,3,2,4,3,3,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,1,3,1,2,0,5),d(2,5,0,5,4,5,0,5,3,5))).
d(97,709,d(d(3,4,2,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,1,4,1,4,a,1),d(2,5,0,5,4,5,0,5,3,5))).
d(99,701,d(d(3,2,2,4,2,4,3,4,5,4,5,5),d(1,2,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,5,3,1,0,4,0,5),d(3,4,0,5,4,5,0,5,5,3))).
d(99,703,d(d(3,4,2,5,2,4,3,4,5,4,5,5),d(1,1,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,3,3,1,4,0,5),d(2,5,0,5,4,5,0,5,5,4))).
d(99,705,d(d(2,4,2,5,5,3,3,3,3,3,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,3,4,0,5,0,5),d(2,5,0,5,2,5,0,5,3,3))).
d(99,707,d(d(2,5,2,5,2,4,3,4,5,3,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,3,1,1,3,0,5),d(2,5,0,5,4,5,0,5,3,5))).
d(99,709,d(d(2,2,2,5,5,4,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,3,2,5,3,3,1,3,0,5),d(2,5,0,5,4,5,0,5,3,5))).

b(101,701,d(d(3,3,2,5,2,5,3,5,5,5,5),d(1,1,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,2,1,0,4,0,5),d(4,5,0,5,4,5,0,5,5,5))).
b(101,703,d(d(3,3,2,5,2,5,3,5,5,5,5),d(5,5,a,5,0,5,0,5,2,4,5,5),d(4,5,4,5,2,0,4,1,2,0,5),d(4,4,0,5,4,5,0,5,5,5))).
b(101,705,d(d(2,5,2,5,2,5,3,4,5,5,5,5),d(5,5,a,5,0,5,0,5,2,3,5,5),d(4,5,4,4,2,0,4,1,4,0,5),d(4,4,0,5,2,5,0,5,5,5))).
b(101,707,d(d(2,4,2,5,2,4,3,3,5,4,5,4),d(5,5,a,5,0,5,0,5,1,4,5,5),d(4,5,1,3,2,3,2,1,3,0,5),d(4,4,0,5,4,5,0,5,5,4))).
b(101,709,d(d(2,2,2,5,4,5,2,4,3,5,5,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,1,4,1,3,4,1,4,0,5),d(2,4,0,5,4,5,0,5,3,4))).
b(103,701,d(d(2,5,2,5,2,3,3,5,5,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,2,1,0,4,0,5),d(4,5,0,5,4,5,0,5,5,5))).
b(103,703,d(d(2,5,2,5,2,5,3,5,5,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,0,5,1,1,0,5),d(4,5,0,5,4,5,0,5,5,5))).
b(103,705,d(d(2,5,2,5,2,4,3,4,5,4,5,5),d(5,5,a,5,0,5,0,5,2,3,5,5),d(4,5,4,4,2,3,2,0,4,0,5),d(4,5,0,5,2,5,0,5,5,5))).
b(103,707,d(d(2,4,2,5,2,3,3,4,5,3,5,5),d(1,1,a,5,0,5,0,5,2,3,5,5),d(4,5,4,3,2,0,5,0,4,0,5),d(4,4,0,5,4,5,0,5,5,5))).
b(103,709,d(d(3,2,2,5,3,2,2,3,3,4,5,4),d(1,1,a,5,0,5,0,5,1,5,5,5),d(4,5,4,3,5,1,1,0,4,0,5),d(2,3,0,5,4,5,0,5,5,2))).
b(105,701,d(d(2,5,2,5,2,4,3,5,5,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,0,5,0,5,0,5),d(4,5,0,5,4,5,0,5,5,5))).
b(105,703,d(d(2,5,2,5,2,5,3,5,5,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,0,4,0,5,0,5),d(4,5,0,5,4,5,0,5,5,5))).
b(105,705,d(d(2,5,2,5,2,5,3,5,5,5,5,5),d(5,5,a,5,0,5,0,5,2,4,5,5),d(4,5,4,5,2,1,1,0,4,0,5),d(4,5,0,5,2,5,0,5,5,5))).
b(105,707,d(d(2,3,2,5,2,4,3,4,5,4,5,5),d(1,1,a,5,0,5,0,5,2,4,5,5),d(4,5,4,4,2,0,5,0,5,0,5),d(4,5,0,5,4,5,0,5,5,5))).
b(105,709,d(d(2,5,2,5,2,5,3,5,5,4,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,0,4,0,4,0,5),d(4,5,0,5,4,5,0,5,4,3))).
b(107,701,d(d(2,5,2,5,2,3,3,5,5,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,0,5,0,5,0,5),d(4,5,0,5,4,5,0,5,5,5))).
b(107,703,d(d(2,5,2,5,2,5,3,5,5,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,4,1,0,5,0,5),d(4,5,0,5,4,5,0,5,5,5))).
b(107,705,d(d(2,5,2,5,2,5,3,5,5,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,0,4,0,4,e,1),d(4,5,0,5,2,5,0,5,5,5))).
b(107,707,d(d(2,5,2,5,2,4,3,4,5,4,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,3,3,4,2,e,3),d(4,5,0,5,4,5,0,5,5,5))).
b(107,709,d(d(2,5,2,5,2,4,3,4,5,4,5,5),d(5,5,a,4,0,5,0,5,2,4,5,5),d(4,5,4,4,2,3,3,4,3,e,3),d(4,5,0,5,4,5,0,5,4,3))).
b(109,701,d(d(2,5,2,5,2,5,3,4,5,4,5,5),d(5,5,a,5,0,5,0,5,1,4,5,5),d(4,5,4,5,2,3,3,4,2,e,3),d(4,5,0,5,4,5,0,5,5,5))).
b(109,703,d(d(2,5,2,5,2,4,3,4,5,4,5,5),d(5,5,a,5,0,5,0,5,1,4,5,5),d(4,5,4,5,2,3,4,4,3,e,4),d(4,5,0,5,4,5,0,5,5,5))).
b(109,705,d(d(2,5,2,5,2,3,2,3,5,3,5,5),d(5,5,a,4,0,5,0,5,2,3,5,5),d(4,5,4,4,2,3,4,0,5,e,4),d(4,5,0,5,2,5,0,5,5,5))).
b(109,707,d(d(2,5,2,5,4,5,2,5,3,5,5,5),d(5,5,a,2,0,5,0,5,2,5,5,5),d(4,5,1,3,2,4,4,4,3,c,4),d(4,3,0,5,4,5,0,5,5,5))).
b(109,709,d(d(2,2,2,5,4,5,2,5,3,5,3,2),d(5,5,a,1,0,5,0,5,2,5,5,5),d(4,5,1,4,2,4,5,2,4,c,5),d(3,3,0,5,4,5,0,5,5,5))).

d(81,691,d(d(4,5,1,5,0,5,0,5,0,5,3,5),d(5,5,0,5,0,5,0,5,1,5,5,5),d(4,5,0,5,0,5,0,5,0,5,0,5),d(3,5,0,5,3,5,0,5,0,5))).
d(83,691,d(d(3,2,1,4,5,4,2,4,3,5,5,3),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,3,1,4,5,4,5,4,5,4),d(4,3,0,5,1,4,0,5,0,5))).
d(83,693,d(d(4,3,1,5,4,3,2,3,3,3,3,2),d(5,5,0,5,2,4,0,5,1,5,5,5),d(4,5,1,3,1,5,5,2,5,2,5,2),d(3,5,0,5,3,3,a,2,0,5))).
d(83,695,d(d(2,5,1,5,4,3,2,3,0,3,5,5),d(5,5,0,5,2,2,0,5,1,5,5,5),d(4,5,1,3,1,5,0,5,0,4,0,5),d(3,4,0,5,3,5,a,4,0,5))).
d(83,697,d(d(2,5,1,5,0,3,0,3,0,5,5,5),d(5,5,0,5,0,5,0,5,1,5,5,5),d(4,5,1,2,1,5,0,5,0,5,e,4),d(3,5,0,5,3,5,a,4,0,5))).
d(83,699,d(d(3,3,1,5,0,5,0,4,0,5,5,5),d(5,5,0,5,0,5,0,5,1,5,5,5),d(4,5,0,5,0,5,0,5,0,5,0,5),d(3,5,0,5,3,5,a,3,0,5))).
d(85,691,d(d(2,4,3,5,5,5,2,5,3,5,5,4),d(5,5,0,4,2,5,0,5,1,5,5,5),d(4,5,2,4,1,5,5,3,5,3,5,3),d(4,5,0,5,1,5,0,5,3,2))).
d(85,693,d(d(3,2,1,5,5,3,2,4,3,5,5,3),d(5,5,0,4,2,3,0,5,1,5,5,5),d(4,5,2,3,5,1,4,1,5,2,5,2),d(4,4,0,5,1,3,0,5,0,4))).
d(85,695,d(d(1,3,1,5,4,3,1,3,3,5,5,5),d(5,5,a,2,2,1,0,5,1,5,5,5),d(4,5,4,3,5,1,3,1,5,1,a,2),d(4,3,0,5,2,3,a,3,3,3))).
d(85,697,d(d(4,5,1,5,4,4,2,4,3,5,5,3),d(1,2,a,5,2,2,0,5,1,5,5,5),d(4,5,1,4,1,5,4,2,5,3,0,5),d(5,3,0,5,1,3,a,4,0,4))).
d(85,699,d(d(4,3,1,5,4,4,2,4,3,4,3,4),d(5,5,a,4,2,4,0,5,1,5,5,5),d(4,5,1,4,1,5,4,3,5,3,a,3),d(5,3,0,5,1,4,a,5,0,5))).
d(87,691,d(d(1,2,4,3,5,3,2,3,3,5,5,4),d(5,5,a,3,2,3,0,5,1,5,5,5),d(4,5,4,3,5,1,4,3,3,2,a,3),d(3,4,0,5,2,4,0,5,3,3))).
d(87,693,d(d(4,3,4,3,4,3,2,3,3,5,3,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,1,3,5,1,5,2,2,2,c,2),d(3,5,0,5,2,3,0,5,3,4))).
d(87,695,d(d(1,2,4,4,4,3,2,3,3,5,5,2),d(5,3,a,5,0,5,0,5,1,3,5,3),d(4,3,1,2,5,1,0,5,0,5,e,2),d(3,2,0,5,2,3,0,5,3,4))).
d(87,697,d(d(2,2,1,2,5,3,2,4,3,5,5,3),d(1,2,a,5,0,5,0,5,1,5,5,5),d(4,5,1,3,2,5,4,2,4,2,a,2),d(2,4,0,5,2,3,0,5,3,5))).
d(87,699,d(d(1,3,4,5,5,4,2,5,3,5,5,5),d(5,5,a,5,0,4,0,5,1,5,5,5),d(4,5,4,4,2,4,1,5,2,4,0,4),d(3,4,0,5,4,5,a,1,3,4))).
d(89,691,d(d(3,2,4,5,4,3,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,3,2,1,4,2,2,2,c,2),d(3,5,0,5,4,5,0,5,3,1))).
d(89,693,d(d(2,4,4,5,5,4,2,5,3,5,3,3),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,4,2,4,4,4,2,4,c,4),d(3,3,0,5,4,5,0,5,3,4))).
d(89,695,d(d(1,3,4,5,4,3,2,3,3,5,5,3),d(5,3,a,4,0,5,0,5,1,4,5,3),d(4,3,1,2,5,1,4,1,0,4,e,2),d(2,3,0,5,2,3,0,5,3,2))).
d(89,697,d(d(2,2,4,3,5,5,2,4,3,5,5,4),d(1,2,a,5,0,5,0,5,1,5,5,5),d(4,5,1,3,2,5,4,2,4,2,c,2),d(2,4,0,5,2,3,0,5,3,5))).
d(89,699,d(d(1,3,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,1,4,2,4,0,5),d(3,5,0,5,4,5,0,5,1,3))).

o(91,691,d(d(2,4,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,3,4,2,5,c,2),d(2,3,0,5,4,5,0,5,3,5))).
o(91,693,d(d(3,3,4,5,5,5,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,4,3,2,4,c,4),d(2,5,0,5,4,5,0,5,3,5))).
o(91,695,d(d(1,3,4,5,5,3,2,4,3,5,5,4),d(5,5,a,3,0,5,0,5,1,5,5,5),d(4,5,4,2,5,4,2,2,1,c,2),d(2,5,0,5,2,4,0,5,3,2))).
o(91,697,d(d(2,3,4,5,5,5,2,4,3,5,5,3),d(1,2,a,5,0,5,0,5,1,5,5,5),d(4,5,1,3,2,4,2,4,2,0,4),d(2,4,0,5,2,3,0,5,1,1))).
o(91,699,d(d(2,5,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,3,5,2,5,0,5),d(3,5,0,5,4,5,0,5,1,5))).
o(93,691,d(d(2,4,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,4,4,2,5,c,4),d(2,5,0,5,4,5,0,5,3,5))).
o(93,693,d(d(2,3,4,5,5,4,2,5,3,5,5,3),d(5,5,a,4,0,5,0,5,1,5,5,5),d(4,5,4,4,2,5,3,5,2,5,0,5),d(2,5,0,5,4,4,0,5,3,5))).
o(93,695,d(d(1,2,4,5,5,3,1,4,3,5,5,3),d(1,1,a,4,0,5,0,5,1,5,5,5),d(4,5,5,4,5,0,4,2,1,a,2),d(2,5,0,5,2,5,0,5,0,5))).
o(93,697,d(d(2,4,4,5,5,5,2,5,3,5,5,4),d(1,1,a,5,0,5,0,5,1,5,5,5),d(4,5,1,3,2,5,3,3,4,1,c,3),d(3,3,0,5,4,4,0,5,0,5))).
o(93,699,d(d(2,4,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,2,3,5,5),d(4,5,4,5,2,5,3,1,0,4,0,5),d(3,5,0,5,4,5,0,5,1,2))).
o(95,691,d(d(3,4,4,5,5,3,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,3,4,2,5,3,4,2,5,c,2),d(2,4,0,5,4,5,0,5,4,3))).
o(95,693,d(d(1,2,4,5,4,3,2,4,3,5,3,3),d(5,5,a,3,0,5,0,5,1,5,5,5),d(4,5,3,3,5,3,3,2,4,a,2),d(2,5,0,5,2,4,0,5,4,2))).
o(95,695,d(d(4,2,4,5,4,3,2,4,3,5,3,3),d(1,2,a,5,0,5,0,5,1,4,5,5),d(4,5,1,4,5,3,2,0,4,a,1),d(2,4,0,5,2,3,0,5,1,2))).
o(95,697,d(d(2,5,4,5,5,4,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,2,4,5,5),d(4,5,4,4,2,5,3,5,2,5,c,2),d(3,4,0,5,4,5,0,5,0,5))).
o(95,699,d(d(2,3,4,5,5,5,2,4,3,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,3,3,4,2,4,0,5),d(3,5,0,5,4,5,0,5,0,5))).
o(97,691,d(d(2,4,4,5,4,4,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,3,4,2,5,3,5,2,5,0,5),d(2,5,0,5,2,2,0,5,4,5))).
o(97,693,d(d(5,3,4,5,4,3,2,3,3,5,5,4),d(5,4,a,3,0,5,a,2,1,5,5,5),d(4,5,5,3,5,3,2,2,3,a,2),d(2,5,0,5,2,5,0,5,4,5))).
o(97,695,d(d(3,3,4,5,5,3,2,5,3,5,5,3),d(1,2,a,5,0,5,0,5,2,3,5,5),d(4,5,4,3,2,5,3,4,2,4,0,5),d(3,3,0,5,4,5,0,5,4,4))).
o(97,697,d(d(2,4,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,5,3,5,2,5,0,5),d(3,5,0,5,4,5,0,5,4,1))).
o(97,699,d(d(2,2,4,5,5,4,1,2,3,4,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,3,3,4,2,5,0,5),d(3,5,0,5,4,5,0,5,4,3))).
o(99,691,d(d(2,4,4,5,4,3,2,5,3,5,3,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,3,4,2,5,3,5,2,5,0,5),d(2,4,0,5,2,3,0,5,4,4))).
o(99,693,d(d(5,2,4,5,5,4,2,3,3,5,5,4),d(5,4,a,3,0,5,0,5,1,5,5,5),d(4,5,4,1,5,3,3,2,3,a,2),d(2,4,0,5,2,5,0,5,4,4))).
o(99,695,d(d(2,5,4,5,5,4,2,5,3,5,5,4),d(1,1,a,5,0,5,0,5,2,4,5,5),d(4,5,4,4,2,5,3,5,2,5,0,5),d(3,3,0,5,4,5,0,5,4,4))).
o(99,697,d(d(2,5,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,5,3,5,2,5,0,5),d(3,5,0,5,4,5,0,5,4,5))).
o(99,699,d(d(1,2,4,5,2,3,3,4,5,3,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,5,3,2,2,3,0,5),d(3,5,0,5,4,5,0,5,4,5))).

o(101,691,d(d(1,3,4,5,4,5,2,5,3,5,3,4),d(5,5,a,5,0,5,i,1,1,5,5,5),d(4,5,3,5,2,
3,5,2,5,0,5),d(3,4,0,5,2,3,0,5,5,4))).
o(101,693,d(d(2,2,4,5,5,4,2,3,3,5,5,4),d(5,4,a,3,0,5,0,5,1,5,5,5),d(4,5,4,2,4,
3,3,2,3,a,2),d(4,4,0,5,2,4,0,5,5,3))).
o(101,695,d(d(2,5,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,2,4,5,5),d(4,5,4,4,2,
3,5,2,5,0,5),d(3,5,0,5,4,5,0,5,5,3))).
o(101,697,d(d(2,3,4,4,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,
3,4,2,5,0,5),d(3,5,0,5,4,5,0,5,4,4))).
o(101,699,d(d(1,3,2,4,2,4,3,4,5,4,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,3,3,
0,4,0,4,0,5),d(3,4,0,5,4,5,0,5,5,3))).
o(103,691,d(d(1,3,4,4,4,4,2,4,3,5,5,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,3,4,4,
3,2,2,3,a,2),d(3,5,0,5,2,4,0,5,5,5))).
o(103,693,d(d(3,4,2,3,5,4,2,4,3,5,5,5),d(1,1,a,4,0,5,0,5,1,5,5,5),d(4,5,4,2,4,
3,3,2,2,a,1),d(4,5,0,5,2,3,0,5,5,4))).
o(103,695,d(d(2,5,2,4,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,2,4,5,5),d(4,5,4,5,2,
3,5,2,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(103,697,d(d(2,3,2,5,5,4,2,4,3,4,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,3,2,
3,3,2,4,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(103,699,d(d(2,3,2,5,2,4,3,4,5,4,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,2,4,3,
0,5,0,5,0,5),d(4,4,0,5,4,5,0,5,5,4))).
o(105,691,d(d(1,3,2,5,5,4,2,3,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,3,4,
3,3,2,2,a,2),d(3,5,0,5,2,4,0,5,5,5))).
o(105,693,d(d(3,5,2,5,5,5,2,5,3,5,5,5),d(5,4,a,5,0,5,0,5,1,5,5,5),d(4,5,4,4,2,
3,5,2,5,0,5),d(4,5,0,5,4,5,0,5,5,4))).
o(105,695,d(d(3,5,2,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,2,3,5,5),d(4,5,4,5,2,
3,5,2,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(105,697,d(d(3,5,2,5,5,3,2,3,3,3,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,2,4,3,
3,3,2,4,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(105,699,d(d(3,3,2,5,2,5,3,5,5,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,2,4,3,
0,5,0,5,0,5),d(4,1,0,5,4,5,0,5,5,5))).
o(107,691,d(d(3,5,2,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,4,2,
3,5,2,4,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(107,693,d(d(3,4,2,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,a,1,1,5,5,5),d(4,5,4,5,2,
3,5,2,5,0,5),d(4,5,0,5,4,5,0,5,5,5))).
o(107,695,d(d(3,3,2,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,2,2,5,5),d(4,5,4,5,2,
3,4,2,4,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(107,697,d(d(2,3,2,5,5,3,2,4,3,4,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,
3,2,2,3,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(107,699,d(d(2,3,2,5,2,5,3,4,5,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,
0,5,0,5,0,5),d(4,3,0,5,4,5,0,5,5,5))).
o(109,691,d(d(2,5,2,4,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,
3,4,2,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(109,693,d(d(2,5,2,3,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,a,3,1,5,5,5),d(4,5,4,5,2,
3,2,2,3,0,5),d(4,5,0,5,4,5,0,5,5,5))).
o(109,695,d(d(2,5,2,3,4,5,2,5,3,5,5,5),d(5,5,a,5,0,5,a,2,2,3,5,5),d(4,5,4,5,2,
3,1,2,1,0,5),d(3,4,0,5,4,5,0,5,5,5))).
o(109,697,d(d(2,5,4,3,4,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,
3,2,4,2,e,2),d(2,3,0,5,4,5,0,5,5,5))).
o(109,699,d(d(2,4,2,5,4,4,2,4,3,4,5,5),d(5,5,a,5,0,5,0,5,2,5,5,5),d(4,5,4,5,2,
3,4,4,3,0,5),d(4,3,0,5,4,5,0,5,5,5))).

a(91,681,d(d(4,3,1,5,5,3,2,3,3,3,3,4),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,3,1,4,5,5,5,4,5,4),d(4,4,0,5,1,5,0,5,0,5))),
a(81,683,d(d(4,4,1,5,0,2,0,2,0,2,3,3),d(5,5,0,5,2,4,0,5,1,5,5,5),d(4,5,2,2,1,3,5,4,5,4,5,3),d(4,3,0,5,1,4,0,5,0,5))),
a(81,685,d(d(4,4,1,5,0,3,0,2,0,2,3,4),d(5,5,0,5,2,4,0,5,1,5,5,5),d(4,5,2,2,1,1,5,3,5,3,5,3),d(3,3,0,5,1,3,0,5,0,5))),
a(81,687,d(d(4,3,1,5,0,4,0,3,0,3,3,3),d(5,5,0,5,2,3,0,5,1,5,5,5),d(4,5,2,1,0,5,2,5,1,5,2),d(3,4,0,5,3,3,0,5,0,5))),
a(81,689,d(d(4,5,1,5,0,5,0,5,0,5,3,5),d(5,5,0,5,3,4,0,5,1,5,5,5),d(4,5,0,5,0,5,0,5,0,5,0,5),d(3,5,0,5,3,4,0,5,0,5))),
a(83,681,d(d(3,3,3,5,5,5,2,5,3,5,5,5),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,5,1,1,4,4,3,3,a,4),d(4,5,0,5,1,5,0,5,0,5))),
a(83,683,d(d(3,4,3,4,5,5,2,5,3,5,5,5),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,5,1,5,5,4,5,4,5,3),d(4,5,0,5,1,5,0,5,0,5))),
a(83,685,d(d(2,2,1,2,5,5,2,5,3,5,5,5),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,5,1,5,5,4,5,5,5,5),d(4,5,0,5,1,4,0,5,1,1))),
a(83,687,d(d(1,3,1,4,5,5,2,5,3,5,3,4),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,5,1,5,5,4,5,4,5,4),d(4,5,0,5,1,4,0,5,1,4))),
a(83,689,d(d(3,3,1,4,5,5,2,5,3,5,5,3),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,5,1,5,5,5,5,4,5,5),d(4,4,0,5,1,5,0,5,1,2))),
a(85,681,d(d(3,2,3,5,5,5,2,5,3,5,5,5),d(5,5,0,5,2,5,0,5,1,5,5,5),d(4,5,2,5,1,5,4,5,3,4,a,4),d(4,5,0,5,1,5,0,5,0,5))),
a(85,683,d(d(1,3,3,5,5,5,2,5,3,5,5,5),d(5,5,0,4,2,5,0,5,1,5,5,5),d(4,5,2,5,1,5,4,5,3,4,a,4),d(4,5,0,5,1,3,0,5,0,5))),
a(85,685,d(d(2,5,3,5,5,5,2,4,3,5,5,5),d(5,5,a,1,2,3,0,5,1,5,5,5),d(4,5,2,4,1,4,4,5,3,3,a,3),d(4,5,0,5,4,5,0,5,0,5))),
a(85,687,d(d(2,4,3,5,5,5,2,5,3,5,5,5),d(5,5,a,1,2,1,0,5,1,5,5,5),d(4,5,4,4,2,3,4,3,4,3,c,4),d(4,4,0,5,4,5,0,5,1,2))),
a(85,689,d(d(2,4,3,5,5,5,2,5,3,5,5,5),d(5,5,0,5,2,4,0,5,1,5,5,5),d(4,5,4,4,2,4,5,3,5,3,5,3),d(4,4,0,5,1,4,0,5,3,3))),
a(87,681,d(d(2,4,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,2,3,0,5,1,5,5,5),d(4,5,4,5,2,4,4,3,2,a,3),d(4,5,0,5,4,5,0,5,0,5))),
a(87,683,d(d(2,4,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,2,2,0,5,1,5,5,5),d(4,5,4,4,2,4,4,3,2,3,c,3),d(4,4,0,5,4,5,0,5,0,5))),
a(87,685,d(d(2,3,3,5,5,5,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,4,2,4,4,3,4,4,c,3),d(3,4,0,5,4,5,0,5,0,5))),
a(87,687,d(d(2,3,3,5,5,4,2,4,3,5,5,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,4,2,4,4,3,4,4,c,4),d(3,5,0,5,2,3,0,5,0,5))),
a(87,689,d(d(2,2,4,3,5,4,1,3,3,5,5,4),d(5,5,a,3,2,3,0,5,1,5,5,5),d(4,5,4,3,5,1,4,2,3,1,a,2),d(3,4,0,5,2,4,0,5,3,1))),
a(89,681,d(d(4,2,3,5,5,4,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,4,2,5,4,3,2,3,c,4),d(4,4,0,5,4,5,0,5,0,4))),
a(89,683,d(d(2,3,3,4,4,3,2,4,3,5,3,3),d(5,5,a,3,0,5,0,5,1,5,5,5),d(4,5,1,3,2,3,4,2,2,2,c,2),d(3,4,0,5,4,4,0,5,0,5))),
a(89,685,d(d(1,3,3,3,4,4,2,4,3,5,5,3),d(5,5,a,3,0,5,0,5,1,5,5,5),d(4,5,1,3,5,0,5,0,5,5,1),d(3,5,0,5,2,4,0,5,0,4))),
a(89,687,d(d(1,3,4,5,5,4,1,4,3,5,5,5),d(5,5,a,2,0,5,0,5,1,5,5,5),d(4,5,4,4,5,4,0,5,0,5,5,2),d(3,5,0,5,2,5,0,5,0,4))),
a(89,689,d(d(4,2,4,5,4,3,2,3,3,5,5,2),d(5,5,a,3,0,5,0,5,1,5,5,5),d(4,5,1,3,5,1,0,4,0,4,a,1),d(3,5,0,5,2,3,0,5,0,4))).

d(91,681,d(d(1,2,4,4,4,4,2,4,3,5,5,3),d(5,5,a,2,0,5,0,5,1,5,5,5),d(4,5,2,4,5,0,4,0,5,5,1),d(3,5,0,5,2,3,0,5,3,3))).
o(91,683,d(d(1,5,4,4,5,4,1,3,3,4,5,2),d(5,4,a,3,0,5,0,5,1,4,5,4),d(4,4,4,2,5,2,2,4,1,5,2),d(3,4,0,5,2,4,0,5,3,3))).
o(91,685,d(d(1,4,4,4,5,4,2,5,3,4,5,3),d(5,4,a,1,0,5,0,5,1,4,5,4),d(4,4,2,4,1,5,2,4,4,2,5,1),d(3,4,0,5,2,4,0,5,3,1))).
o(91,687,d(d(5,2,4,3,5,3,2,5,3,5,3,3),d(1,3,a,4,0,5,0,5,1,3,5,3),d(4,3,4,2,5,0,4,0,4,e,2),d(3,3,0,5,0,3,0,5,3,2))).
o(91,689,d(d(2,4,4,5,5,3,2,5,3,5,5,4),d(5,4,a,5,0,5,0,5,1,5,5,5),d(4,5,4,4,2,4,4,4,2,4,c,4),d(3,5,0,5,4,5,0,5,3,4))).
o(93,681,d(d(2,3,4,3,5,4,2,3,3,4,5,3),d(5,4,a,4,0,5,0,5,1,3,5,4),d(4,4,2,2,5,1,2,3,4,2,5,1),d(3,3,0,5,2,4,0,5,5,3))).
o(93,683,d(d(1,3,4,4,5,5,1,3,3,5,5,4),d(4,4,a,3,0,5,0,5,1,5,5,4),d(4,5,4,3,5,0,2,2,3,2,5,1),d(3,5,0,5,2,4,0,5,5,5))).
o(93,685,d(d(4,2,4,5,4,4,1,3,3,5,5,5),d(1,2,a,4,0,5,0,5,1,5,5,5),d(4,5,4,3,5,0,4,2,1,5,2),d(3,5,0,5,2,3,0,5,5,4))).
o(93,687,d(d(2,3,4,5,4,4,2,5,3,5,3,3),d(1,2,a,5,0,5,0,5,1,5,5,4),d(4,5,1,3,1,5,4,1,2,4,c,2),d(3,5,0,5,4,4,0,5,5,4))).
o(93,689,d(d(2,5,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,4,5,2,5,c,4),d(3,4,0,5,4,5,0,5,3,4))).
o(95,681,d(d(3,2,4,5,5,5,2,4,3,5,3,3),d(5,4,a,3,0,5,0,5,1,5,5,5),d(4,5,2,4,5,0,2,3,3,4,5,1),d(3,4,0,5,2,4,0,5,5,5))).
o(95,683,d(d(1,3,4,5,5,4,1,4,3,5,5,4),d(5,5,a,4,0,5,0,5,1,5,5,5),d(4,5,4,3,5,0,0,5,0,5,f,2),d(3,5,0,5,2,4,0,5,5,4))).
o(95,685,d(d(3,2,4,5,5,4,2,5,3,5,5,3),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,1,3,1,5,3,3,2,4,0,5),d(3,5,0,5,4,4,0,5,3,4))).
o(95,687,d(d(2,4,4,5,5,4,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,4,2,4,3,4,2,5,0,5),d(3,5,0,5,4,5,0,5,5,3))).
o(95,689,d(d(2,4,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,4,2,5,3,4,2,5,c,1),d(3,5,0,5,4,5,0,5,5,4))).
o(97,681,d(d(1,3,4,5,5,5,1,4,3,4,5,4),d(5,5,a,4,0,5,0,5,1,5,5,5),d(4,5,4,4,5,0,0,5,0,4,a,2),d(3,5,0,5,2,5,0,5,5,5))).
o(97,683,d(d(4,3,4,5,4,4,2,4,3,5,3,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,1,4,1,5,0,5,0,5,0,4),d(3,5,0,5,2,3,0,5,5,4))).
o(97,685,d(d(2,5,4,5,5,5,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,4,3,4,2,4,0,5),d(3,5,0,5,4,5,0,5,5,3))).
o(97,687,d(d(2,3,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,3,5,2,5,0,5),d(3,5,0,5,4,5,0,5,3,4))).
o(97,689,d(d(2,4,4,5,5,4,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,4,2,5,3,5,2,5,0,5),d(3,5,0,5,4,5,0,5,4,3))).
o(99,681,d(d(1,3,4,4,5,5,1,5,3,4,5,4),d(5,4,a,5,0,5,0,5,1,4,5,4),d(4,3,4,4,5,0,0,5,0,5,a,2),d(3,4,0,5,2,4,0,5,5,5))).
o(99,683,d(d(2,5,4,2,5,5,0,4,0,4,5,2),d(0,3,a,5,0,5,0,5,1,2,5,2),d(4,2,1,2,2,0,0,5,0,5,0,5),d(3,2,0,5,0,4,0,5,5,5))).
o(99,685,d(d(2,5,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,3,3,2,3,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(99,687,d(d(2,5,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,5,3,4,2,4,0,5),d(3,5,0,5,4,5,0,5,5,3))).
o(99,689,d(d(2,5,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,4,2,5,3,5,2,5,0,5),d(3,5,0,5,4,5,0,5,3,3))).

o(101,681,d(d(1,4,4,4,5,5,1,5,3,4,5,4),d(5,4,a,5,0,5,0,5,1,4,5,4),d(4,3,4,4,5,0,5,0,5,f,3),d(3,3,0,5,2,4,0,5,5,5))).
o(101,683,d(d(2,5,4,2,5,5,0,3,0,4,5,2),d(0,3,a,5,0,5,0,5,1,2,5,2),d(4,3,4,2,2,0,5,0,5,0,5),d(2,2,0,5,0,4,0,5,5,5))).
o(101,685,d(d(2,4,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,3,2,2,3,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(101,687,d(d(2,5,4,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,3,4,2,4,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(101,689,d(d(2,4,4,5,4,4,2,5,3,5,5,4),d(5,5,a,5,0,5,i,1,1,5,5,5),d(4,5,3,4,2,3,5,2,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(103,681,d(d(3,5,4,3,5,5,1,5,3,3,5,3),d(5,4,a,5,0,5,0,5,1,3,5,3),d(4,3,4,2,2,0,5,0,5,f,3),d(4,2,0,5,2,3,0,5,5,1))).
o(103,685,d(d(1,5,4,2,4,5,0,4,0,4,5,3),d(5,2,a,5,0,5,0,5,1,2,5,3),d(4,2,4,2,2,0,5,0,5,0,5),d(2,2,0,5,0,4,0,5,5,5))).
o(103,687,d(d(2,3,4,5,4,5,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,3,5,2,3,5,2,4,0,5),d(3,4,0,5,4,4,0,5,5,5))).
o(103,689,d(d(1,3,4,4,4,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,3,5,2,3,4,2,4,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(105,681,d(d(3,3,4,1,0,4,0,4,0,4,0,5),d(0,4,0,4,0,5,0,5,0,4,0,4),d(0,4,0,4,0,0,5,0,5,0,4),d(0,4,0,5,0,4,0,5,5,2))).
o(105,683,d(d(1,3,4,1,0,4,0,4,0,5,0,4),d(0,4,a,5,0,5,0,5,1,2,0,4),d(4,2,4,1,2,4,2,0,5,c,2),d(0,4,0,5,0,4,0,5,5,5))).
o(105,685,d(d(2,3,4,2,4,5,2,4,3,4,5,3),d(5,3,a,5,0,5,0,5,1,3,5,3),d(4,3,3,3,2,4,2,0,5,c,2),d(3,2,0,5,0,3,0,5,5,5))).
o(105,687,d(d(2,3,2,4,4,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,3,5,2,3,4,2,4,0,5),d(3,4,0,5,4,3,0,5,5,5))).
o(105,689,d(d(1,5,2,5,4,4,2,4,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,3,4,2,3,3,2,4,a,1),d(3,5,0,5,2,4,0,5,5,5))).
o(107,681,d(d(2,4,4,2,5,5,0,4,0,4,0,4),d(0,4,0,4,0,5,0,5,0,4,0,4),d(0,4,0,4,0,0,5,0,5,0,5),d(0,4,0,5,0,4,0,5,5,4))).
o(107,683,d(d(2,4,4,3,4,5,2,5,0,4,5,3),d(0,3,a,5,0,5,0,5,1,5,0,4),d(4,3,3,2,2,0,5,0,5,0,5),d(3,2,0,5,0,3,0,5,5,5))).
o(107,685,d(d(2,3,3,3,5,4,2,4,3,5,5,5),d(5,5,a,4,0,5,0,5,1,5,5,5),d(4,5,3,4,4,4,1,0,5,a,1),d(3,5,0,5,2,3,0,5,5,5))).
o(107,687,d(d(1,4,2,5,5,4,2,3,3,5,5,5),d(5,5,a,4,0,5,0,5,1,5,5,5),d(4,5,4,3,4,0,5,0,5,a,2),d(3,5,0,5,2,4,0,5,5,5))).
o(107,689,d(d(3,5,2,5,5,4,2,4,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,3,4,3,3,2,1,a,1),d(3,5,0,5,2,2,0,5,5,5))).
o(109,681,d(d(1,3,4,5,4,3,2,3,3,5,5,5),d(5,5,a,5,0,5,0,5,0,5,4,4),d(2,5,4,3,2,3,2,3,2,0,5),d(3,4,0,5,4,4,0,5,5,5))).
o(109,683,d(d(1,5,4,5,4,4,2,4,3,5,5,5),d(5,5,a,4,0,5,0,5,1,5,5,3),d(2,3,3,4,2,0,5,0,5,a,1),d(3,4,0,5,2,4,0,5,5,5))).
o(109,685,d(d(3,3,4,3,5,5,1,4,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,3,4,0,5,0,5,a,2),d(3,4,0,5,2,4,0,5,5,5))).
o(109,687,d(d(2,3,2,4,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,2,3,2,3,4,2,2,0,5),d(3,5,0,5,4,4,0,5,5,5))).
o(109,689,d(d(2,5,2,4,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,5,5,5),d(4,5,4,5,2,3,5,2,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).

o(91,641,d(d(2,4,3,3,2,5,3,3,5,5,5,3),d(5,3,0,5,0,5,0,5,4,3,0,3),d(1,3,3,3,4,5,0,5,0,5,0,5),d(3,3,0,5,4,3,0,5,5,5))).
o(91,643,d(d(4,4,3,5,2,4,3,4,5,5,3,3),d(5,5,0,5,0,5,0,5,4,5,3,5),d(1,5,3,4,4,4,2,1,2,1,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(91,645,d(d(4,4,3,5,3,3,4,4,5,5,3,3),d(5,5,0,5,0,5,0,5,4,5,3,5),d(1,5,1,4,2,4,2,4,2,4,0,5),d(3,5,0,5,4,5,0,5,5,4))).
o(91,647,d(d(2,3,3,5,3,3,4,5,5,5,2,3),d(5,5,0,5,0,5,0,5,4,5,3,3),d(1,4,1,5,2,4,2,4,2,4,0,5),d(3,5,0,5,4,5,0,5,3,1))).
o(91,649,d(d(3,2,3,5,4,3,5,4,5,5,3,3),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,2,3,1,2,2,4,2,4,0,5),d(3,5,0,5,4,5,0,5,3,3))).
o(93,641,d(d(2,4,3,5,2,5,3,5,5,5,5,5),d(5,4,0,5,0,5,0,5,3,4,3,4),d(1,5,3,5,4,5,0,5,0,5,0,5),d(3,5,0,5,4,4,0,5,5,5))).
o(93,643,d(d(2,5,3,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,4,4,3,5),d(1,5,3,5,4,5,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(93,645,d(d(2,4,3,5,2,4,3,5,5,5,5,4),d(5,5,0,5,0,5,0,5,4,5,3,5),d(1,5,3,4,4,2,1,0,4,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(93,647,d(d(2,3,3,5,3,4,4,4,5,5,2,2),d(5,5,0,5,0,5,0,5,4,5,3,5),d(1,5,1,4,2,2,4,2,4,0,5),d(3,5,0,5,4,5,0,5,5,2))).
o(93,649,d(d(2,3,3,5,4,4,5,4,5,5,2,3),d(5,5,0,5,0,5,0,5,4,5,3,3),d(1,3,2,3,1,2,3,2,3,0,5),d(3,5,0,5,4,5,0,5,0,4))).
o(95,641,d(d(4,3,3,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,4),d(1,5,3,5,4,5,0,5,0,5,0,5),d(3,5,0,5,4,4,0,5,5,4))).
o(95,643,d(d(2,3,3,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,4,3,5),d(1,5,3,5,4,5,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(95,645,d(d(1,4,3,5,2,4,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,4,5,3,5),d(1,5,3,5,4,5,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(95,647,d(d(2,2,3,5,2,4,3,4,5,5,3,3),d(5,5,0,5,0,5,0,5,4,5,3,5),d(1,5,3,4,4,2,2,2,2,2,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(95,649,d(d(2,4,3,5,3,4,4,4,5,5,2,3),d(5,5,0,5,0,5,0,5,4,5,3,5),d(1,5,1,5,2,2,4,2,4,0,5),d(3,5,0,5,4,5,0,5,5,3))).
o(97,641,d(d(2,3,3,3,2,4,3,3,5,5,0,3),d(5,2,0,5,0,5,0,5,3,2,0,3),d(1,3,3,3,4,2,1,0,4,0,5),d(3,3,0,5,0,3,0,5,5,3))).
o(97,643,d(d(4,4,3,5,2,5,3,5,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,4,5,0,5,0,5,0,4),d(3,5,0,5,4,5,0,5,5,5))).
o(97,645,d(d(2,5,3,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,4,4,3,5),d(1,5,3,5,4,5,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(97,647,d(d(1,4,3,5,2,5,3,5,5,5,5,4),d(5,5,0,5,0,5,0,5,4,4,3,5),d(1,5,3,5,4,5,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(97,649,d(d(2,3,3,5,2,3,3,3,5,5,5,3),d(5,5,0,5,0,5,0,5,3,4,3,5),d(1,5,1,3,2,2,3,2,3,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(99,641,d(d(3,4,4,4,4,3,5,3,5,5,3,3),d(5,5,0,5,0,5,0,5,3,5,3,4),d(1,5,4,3,4,2,5,2,5,0,5),d(3,5,0,5,4,5,0,5,5,4))).
o(99,643,d(d(2,4,4,4,2,5,5,3,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,4,4,2,2,2,2,2,0,5),d(3,5,0,5,4,5,0,5,5,3))).
o(99,645,d(d(4,3,4,4,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,4,5,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(99,647,d(d(1,5,4,4,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,5,4,5,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5))).
o(99,649,d(d(4,2,4,5,2,4,3,4,5,5,5,3),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,4,4,2,2,2,2,2,0,5),d(3,4,0,5,4,5,0,5,5,5))).

o(91,671,d(d(2,5,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,4,4,5),d(2,5,4,5,2,
2,4,4,4,0,5),d(4,5,0,5,4,5,0,5,2,5))).
o(91,673,d(d(2,5,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,2,4,5,5),d(4,4,4,5,2,
4,4,3,3,c,3),d(4,5,0,5,4,5,0,5,2,5))).
o(91,675,d(d(2,5,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,2,5,5),d(4,5,4,5,2,
4,4,3,3,c,3),d(4,5,0,5,4,5,0,5,1,5))).
o(91,677,d(d(2,2,3,5,5,4,2,5,3,5,5,4),d(5,5,a,4,0,5,0,5,1,5,5,5),d(4,5,4,4,2,
4,2,3,2,c,2),d(4,5,0,5,4,5,0,5,1,3))).
o(91,679,d(d(4,2,3,4,4,4,2,3,3,5,3,3),d(5,5,a,3,0,5,0,5,1,5,5,5),d(4,5,2,4,2,
0,4,0,5,0,5),d(4,3,0,5,4,4,0,5,3,2))).
o(93,671,d(d(2,4,3,5,4,4,2,4,3,5,5,5),d(5,5,a,2,0,5,0,5,3,5,5,3),d(2,3,2,4,2,
2,4,4,4,5,1),d(4,5,0,5,4,5,0,5,2,4))).
o(93,673,d(d(2,3,4,2,4,4,2,4,3,5,5,3),d(5,5,0,4,0,5,0,5,3,5,5,5),d(4,4,2,4,2,
2,4,4,4,5,1),d(4,5,0,5,4,5,0,5,2,3))).
o(93,675,d(d(2,4,4,3,4,4,2,4,3,5,5,3),d(5,4,a,1,0,5,0,5,3,4,5,4),d(4,4,2,5,2,
2,4,4,3,5,2),d(4,4,0,5,4,4,0,5,1,1))).
o(93,677,d(d(2,4,4,4,4,4,2,4,3,5,5,2),d(5,4,a,1,0,5,0,5,3,4,5,4),d(4,4,2,4,2,
2,4,4,4,5,2),d(4,3,0,5,4,4,0,5,3,1))).
o(93,679,d(d(3,3,4,4,4,4,2,4,3,5,3,2),d(5,4,a,2,0,5,0,5,3,3,5,5),d(4,4,2,4,2,
2,4,4,4,5,1),d(3,2,0,5,2,4,0,5,5,1))).
o(95,671,d(d(3,3,4,5,4,5,2,5,3,5,3,3),d(5,5,a,5,0,5,0,5,3,1,5,3),d(2,3,2,4,2,
2,3,4,5,c,1),d(4,5,0,5,4,5,0,5,5,5))).
o(95,673,d(d(2,3,4,3,4,5,2,3,3,3,5,2),d(5,4,a,5,0,5,0,5,3,2,5,3),d(4,3,2,2,2,
4,2,4,1,c,2),d(4,3,0,5,4,4,0,5,5,5))).
o(95,675,d(d(2,5,4,1,0,5,0,5,0,5,0,5),d(0,5,0,5,0,5,0,5,0,5,0,5),d(0,5,0,5,0,
0,5,0,5,0,5),d(0,5,0,5,0,4,0,5,5,3))).
o(95,677,d(d(3,5,4,1,0,5,0,5,0,5,0,5),d(0,5,a,5,0,5,0,5,0,5,0,5),d(0,5,0,5,0,
0,5,0,5,0,5),d(0,5,0,5,0,5,0,5,5,3))).
o(95,679,d(d(3,3,4,3,5,5,0,4,0,4,0,4),d(0,4,a,4,0,5,0,5,1,2,0,4),d(4,2,2,3,2,
2,2,0,4,0,5),d(3,1,0,5,0,4,0,5,5,5))).
o(97,671,d(d(3,3,4,5,4,5,2,5,3,5,3,3),d(5,5,a,5,0,5,0,5,3,4,4,4),d(2,4,2,5,2,
3,4,1,4,c,2),d(4,5,0,5,4,5,0,5,5,3))).
o(97,673,d(d(1,4,4,5,4,4,2,3,3,5,5,5),d(5,5,a,5,0,5,0,5,1,3,5,5),d(4,5,2,3,5,
3,3,1,4,c,2),d(4,5,0,5,4,5,0,5,5,5))).
o(97,675,d(d(1,3,4,4,5,4,1,4,3,4,5,4),d(5,4,a,3,0,5,0,5,1,4,5,4),d(4,4,4,3,5,
4,4,1,4,c,4),d(4,4,0,5,2,2,0,5,5,5))).
o(97,677,d(d(1,2,4,5,5,5,1,3,3,5,5,4),d(5,4,a,3,0,5,0,5,1,5,5,4),d(4,5,4,3,5,
2,3,3,3,5,1),d(4,4,0,5,2,5,0,5,5,5))).
o(97,679,d(d(2,3,4,5,5,4,2,3,3,5,5,3),d(5,5,a,4,0,5,0,5,1,5,5,5),d(4,5,4,3,5,
3,3,2,1,5,2),d(4,5,0,5,2,4,0,5,5,5))).
o(99,671,d(d(3,5,4,5,4,5,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,3,4,5,3),d(4,3,2,5,2,
3,3,1,2,c,2),d(4,5,0,5,4,5,0,5,4,5))).
o(99,673,d(d(3,3,4,5,4,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,1,3,5,5),d(4,5,2,5,2,
4,4,1,5,c,4),d(4,5,0,5,4,4,0,5,4,3))).
o(99,675,d(d(2,3,4,5,4,4,2,4,3,5,5,4),d(5,5,a,4,0,5,0,5,1,5,5,5),d(4,5,2,3,5,
4,2,1,2,c,2),d(4,5,0,5,2,3,0,5,5,2))).
o(99,677,d(d(2,4,4,5,5,5,2,3,3,5,5,4),d(5,5,a,4,0,5,0,5,1,5,5,5),d(4,5,2,4,5,
3,4,4,2,5,1),d(4,4,0,5,2,3,0,5,5,3))).
o(99,679,d(d(1,4,4,5,5,4,2,4,3,4,3,3),d(5,5,a,3,0,5,0,5,1,5,5,5),d(4,5,2,3,2,
3,4,4,3,0,4),d(4,3,0,5,2,3,0,5,5,5))).

o(91,661,d(d(2,4,3,5,4,4,2,4,3,4,5,5),d(5,5,a,2,0,5,0,5,3,5,4,5),d(2,5,4,5,2,4,4,1,4,1,0,4),d(4,5,0,5,4,5,0,5,3,5))).
o(91,663,d(d(2,5,3,5,4,3,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,5,2,5,4,2,4,1,c,2),d(4,5,0,5,4,5,0,5,3,5))).
o(91,665,d(d(2,4,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,5,2,5,4,2,4,1,c,2),d(4,5,0,5,4,5,0,5,4,3))).
o(91,667,d(d(2,3,3,5,5,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,5,2,5,2,2,4,2,0,5),d(4,5,0,5,4,5,0,5,4,5))).
o(91,669,d(d(2,3,3,3,5,5,2,3,3,5,5,4),d(5,4,a,5,0,5,0,5,3,3,4,5),d(2,3,4,5,2,5,2,3,4,4,0,5),d(4,3,0,5,4,4,0,5,5,2))).
o(93,661,d(d(1,4,3,5,2,4,4,4,5,4,5,5),d(5,5,0,5,0,5,0,5,3,5,4,4),d(2,3,4,4,4,4,0,5,0,5,0,5),d(4,5,0,5,4,5,0,5,3,5))).
o(93,663,d(d(1,5,3,5,4,5,2,5,3,5,5,5),d(5,5,0,4,0,5,0,5,3,5,4,5),d(2,5,4,5,2,5,4,2,4,3,c,2),d(4,5,0,5,4,5,0,5,3,5))).
o(93,665,d(d(1,5,3,5,4,5,2,4,3,5,5,5),d(5,5,a,2,0,5,0,5,3,5,4,5),d(2,5,4,5,2,5,4,2,4,2,c,2),d(4,5,0,5,4,5,0,5,3,4))).
o(93,667,d(d(1,5,3,5,4,5,2,4,3,5,5,5),d(5,5,a,3,0,5,0,5,3,5,4,5),d(2,5,2,4,2,5,2,4,4,4,5,1),d(4,5,0,5,4,5,0,5,5,4))).
o(93,669,d(d(2,3,3,5,4,5,2,4,3,5,5,5),d(5,5,a,3,0,5,0,5,3,5,5,3),d(2,4,2,4,2,5,2,5,4,5,5,1),d(4,5,0,5,4,5,0,5,5,3))).
o(95,661,d(d(1,5,3,5,2,5,4,5,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,4,4),d(2,4,1,3,2,5,0,5,0,5,a,1),d(4,5,0,5,4,5,0,5,3,5))).
o(95,663,d(d(1,5,3,5,4,4,2,4,3,4,5,5),d(5,5,a,3,0,5,0,5,3,5,4,5),d(2,5,4,4,2,5,3,2,1,1,a,2),d(4,5,0,5,4,5,0,5,3,5))).
o(95,665,d(d(3,3,3,4,4,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,5,2,5,3,3,1,3,5,1),d(4,5,0,5,4,5,0,5,3,5))).
o(95,667,d(d(1,5,4,3,4,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,5,2,5,3,3,4,1,0,5),d(4,5,0,5,4,5,0,5,5,5))).
o(95,669,d(d(2,3,4,4,4,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,5,3),d(2,5,4,4,2,5,2,3,4,4,0,5),d(4,5,0,5,4,5,0,5,5,5))).
o(97,661,d(d(1,5,4,4,2,5,4,5,5,5,5,5),d(5,5,0,4,0,5,0,5,3,5,4,4),d(2,4,3,4,3,5,0,5,0,5,a,1),d(4,5,0,5,4,5,0,5,3,5))).
o(97,663,d(d(3,2,4,5,5,3,1,3,3,4,5,5),d(5,5,a,4,0,5,0,5,3,5,4,5),d(2,5,5,4,5,0,0,4,0,5,a,1),d(4,5,0,5,4,5,0,5,3,5))).
o(97,665,d(d(1,3,4,5,5,3,2,3,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,3,5,0,3,3,2,2,0,5),d(4,5,0,5,4,5,0,5,3,5))).
o(97,667,d(d(1,4,4,5,4,5,2,5,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,4,4,2,5,3,3,2,2,0,5),d(4,5,0,5,4,5,0,5,5,5))).
o(97,669,d(d(3,2,4,5,4,5,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,3,5,4,4),d(2,5,2,4,2,5,3,3,1,2,0,5),d(4,5,0,5,4,5,0,5,5,5))).
o(99,661,d(d(3,4,4,5,2,5,4,5,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,4,3),d(2,3,3,4,3,5,0,5,0,5,a,1),d(4,5,0,5,4,5,0,5,3,3))).
o(99,663,d(d(3,2,4,5,5,4,1,4,3,5,5,5),d(5,5,a,4,0,5,0,5,3,5,4,5),d(2,5,5,4,5,4,3,3,2,2,0,5),d(4,5,0,5,4,5,0,5,3,5))).
o(99,665,d(d(1,4,4,5,4,5,2,4,3,5,5,5),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,2,9,2,5,3,3,2,3,0,5),d(4,5,0,5,4,5,0,5,3,5))).
o(99,667,d(d(1,4,4,5,4,5,2,5,3,5,5,4),d(5,5,a,5,0,5,0,5,3,5,4,5),d(2,5,2,4,2,5,3,2,2,1,0,5),d(4,5,0,5,4,5,0,5,5,4))).
o(99,669,d(d(3,3,4,5,4,5,2,5,3,5,3,5),d(5,5,a,5,0,5,0,5,3,5,5,3),d(2,5,2,5,2,5,3,1,2,1,0,5),d(4,5,0,5,4,5,0,5,5,5))).

a(91,651,d(d(2,3,3,5,2,5,3,5,5,5,2,3),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,3,2,5,4,4,0,5),d(3,5,0,5,4,5,0,5,0,5))).
a(91,653,d(d(1,3,3,5,2,5,4,4,5,5,3,2),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,5,2,2,4,2,3,0,5),d(3,5,0,5,4,5,0,5,3,2))).
a(91,655,d(d(3,2,3,5,2,5,4,5,5,5,5,4),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,4,2,1,2,1,2,0,5),d(3,5,0,5,4,5,0,5,3,4))).
a(91,657,d(d(3,4,3,5,2,5,4,5,5,5,5,5),d(5,5,0,5,0,5,0,5,4,3,2,5),d(3,5,4,3,4,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,3,5))).
a(91,659,d(d(1,4,3,5,2,4,4,4,5,4,5,5),d(5,5,0,5,0,5,0,5,3,5,2,4),d(3,4,2,3,4,0,5,0,5,0,5),d(3,4,0,5,4,5,0,5,3,5))).
a(93,651,d(d(2,3,3,4,2,4,4,3,5,5,3,3),d(5,5,0,5,0,5,0,5,4,5,2,5),d(3,5,1,4,2,2,4,4,4,0,5),d(3,5,0,5,4,5,0,5,4))).
a(93,653,d(d(3,2,3,5,2,5,4,5,5,5,3,3),d(5,5,0,5,0,5,0,5,4,4,2,5),d(3,5,1,5,2,2,2,1,2,0,5),d(3,5,0,5,4,5,0,5,3,5))).
a(93,655,d(d(3,2,3,5,2,5,4,5,5,5,5,4),d(5,5,0,5,0,5,0,5,3,4,2,5),d(3,5,1,5,2,1,3,1,1,0,5),d(3,5,0,5,4,5,0,5,3,5))).
a(93,657,d(d(2,3,3,5,2,5,4,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,4,4,4,0,4,0,4,0,5),d(3,5,0,5,4,5,0,5,3,5))).
a(93,659,d(d(2,4,3,5,2,5,4,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,4,5,4,0,5,0,5,0,5),d(4,4,0,5,4,5,0,5,3,5))).
a(95,651,d(d(3,3,3,5,4,4,5,4,5,5,3,3),d(5,5,0,5,0,5,0,5,4,3,2,4),d(3,4,2,3,1,2,2,4,3,0,5),d(3,5,0,5,4,5,0,5,0,4))).
a(95,653,d(d(2,3,3,5,4,3,5,3,5,5,3,2),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,2,0,4,0,4,0,5),d(3,5,0,5,4,5,0,5,3,1))).
a(95,655,d(d(2,3,3,5,2,5,4,5,5,5,5,3),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,2,1,2,3,2,0,5),d(3,5,0,5,4,5,0,5,3,2))).
a(95,657,d(d(3,2,3,5,2,5,4,5,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,4,4,2,2,3,2,0,5),d(3,3,0,5,4,5,0,5,3,3))).
a(95,659,d(d(1,3,3,5,2,5,4,5,5,5,3,4),d(5,5,0,5,0,5,0,5,3,5,2,4),d(3,4,4,3,4,0,5,0,5,0,5),d(4,5,0,5,4,5,0,5,3,5))).
a(97,651,d(d(3,4,4,4,3,2,4,4,5,5,2,4),d(5,5,0,5,0,5,0,5,3,5,3,4),d(1,4,1,4,2,2,3,4,3,0,5),d(3,5,0,5,4,5,0,5,5,2))).
a(97,653,d(d(3,3,3,5,4,5,5,5,5,5,2,3),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,3,1,2,2,2,2,0,5),d(3,5,0,5,4,5,0,5,0,5))).
a(97,655,d(d(3,3,3,5,2,5,4,4,5,5,3,3),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,1,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,1,1))).
a(97,657,d(d(3,3,3,4,2,5,4,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,1,3,1,0,4,0,5),d(4,5,0,5,4,5,0,5,1,3))).
a(97,659,d(d(1,4,4,3,2,5,4,5,5,5,3,4),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,3,3,0,4,2,1,0,5),d(4,5,0,5,4,5,0,5,3,3))).
a(99,651,d(d(2,4,4,5,3,4,4,5,5,5,2,4),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,1,4,2,2,4,4,5,0,5),d(4,4,0,5,4,5,0,5,5,1))).
a(99,653,d(d(2,3,4,5,4,3,4,3,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,2,3),d(3,4,2,3,1,2,2,2,3,0,5),d(4,5,0,5,4,5,0,5,0,5))).
a(99,655,d(d(2,2,4,5,2,4,4,4,5,5,2,3),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,1,2,1,2,1,0,5),d(4,5,0,5,4,5,0,5,1,2))).
a(99,657,d(d(1,3,4,5,2,5,4,5,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,5,1,0,5,0,5,0,5),d(4,5,0,5,4,5,0,5,1,4))).
a(99,659,d(d(3,3,4,5,2,5,4,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,2,5),d(3,5,1,4,3,3,1,2,1,0,4),d(4,5,0,5,4,5,0,5,1,5))).

o(91,631,d(d(2,4,3,5,4,5,5,5,5,5,3),d(5,5,0,5,f,4,a,4,5,5,3,5),d(0,4,2,4,1,4,3,4,2,4,e,5),d(4,5,0,5,4,3,0,5,5,3)))
o(91,633,d(d(2,2,3,2,4,5,0,3,0,3,5,4),d(5,3,0,5,0,5,a,4,3,3,0,3),d(1,3,4,3,0,4,3,4,2,3,e,3),d(4,2,0,5,4,3,0,5,5,5)))
o(91,635,d(d(3,2,3,5,4,4,5,4,5,5,5,3),d(5,5,0,5,0,5,0,4,3,5,3,5),d(1,5,3,4,1,4,3,4,2,4,e,4),d(3,5,0,5,4,4,0,5,5,5)))
o(91,637,d(d(1,3,3,5,2,4,3,4,5,5,5,4),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,4,4,5,3,2,2,1,a,1),d(3,5,0,5,4,5,0,5,5,5)))
o(91,639,d(d(2,4,3,5,2,5,3,5,5,5,5,5),d(5,5,0,5,0,5,0,5,3,4,3,5),d(1,5,3,5,4,5,0,5,0,5,0,5),d(3,5,0,5,4,5,0,5,5,5)))
o(93,631,d(d(4,3,4,4,4,5,5,5,5,5,1,4),d(5,5,0,5,f,5,a,3,5,5,3,5),d(0,4,2,5,1,5,4,3,3,4,a,3),d(4,5,0,5,2,5,0,5,0,5)))
o(93,633,d(d(2,3,3,5,4,5,5,5,5,5,5,3),d(5,5,0,5,3,3,a,4,5,2,3,5),d(1,5,4,3,1,5,3,5,2,5,e,5),d(4,4,0,5,4,3,0,5,5,3)))
o(93,635,d(d(2,4,3,5,4,5,5,5,5,5,5,5),d(5,5,0,5,0,5,a,2,4,1,3,5),d(1,5,4,5,1,5,3,5,2,5,e,5),d(5,3,0,5,4,5,0,5,5,5)))
o(93,637,d(d(1,3,3,5,4,4,5,4,5,5,5,4),d(5,5,0,5,0,5,a,3,3,5,3,5),d(1,5,4,4,1,4,3,4,2,4,e,4),d(3,5,0,5,4,5,0,5,5,5)))
o(93,639,d(d(1,4,3,5,2,4,3,4,5,5,5,5),d(5,5,0,5,0,5,0,5,3,5,3,5),d(1,5,3,4,4,5,3,2,2,1,a,1),d(3,5,0,5,4,5,0,5,5,5)))
o(95,631,d(d(4,3,4,5,4,5,5,5,5,5,1,5),d(5,5,0,5,f,1,0,5,5,5,3,5),d(0,4,2,5,1,5,4,5,3,5,a,4),d(4,5,0,5,2,5,0,5,0,5)))
o(95,633,d(d(4,3,4,4,4,5,5,5,5,5,1,4),d(5,5,0,5,3,5,0,5,5,5,3,5),d(1,5,2,4,1,5,4,5,3,3,a,3),d(4,5,0,5,2,5,0,5,0,5)))
o(95,635,d(d(2,5,3,4,4,5,5,5,5,5,5,4),d(5,5,0,5,3,2,a,5,5,4,3,5),d(1,5,4,4,1,5,3,5,2,5,e,5),d(5,3,0,5,4,4,0,5,5,4)))
o(95,637,d(d(2,4,3,4,4,5,5,5,5,5,5,4),d(5,5,0,5,0,5,a,4,4,1,3,5),d(1,5,4,5,1,5,3,5,2,5,e,5),d(3,3,0,5,4,5,0,5,5,5)))
o(95,639,d(d(2,3,3,4,4,4,5,4,5,5,3,2),d(5,4,0,5,0,5,0,5,3,4,3,4),d(1,4,4,3,4,2,3,3,2,4,e,3),d(3,4,0,5,4,4,0,5,5,3)))
o(97,631,d(d(4,4,4,5,4,5,5,5,5,5,1,5),d(5,5,0,5,3,5,0,5,5,5,3,5),d(0,4,2,3,1,5,4,5,3,5,a,3),d(4,5,0,5,2,5,0,5,0,5)))
o(97,633,d(d(4,3,4,5,4,5,5,5,5,5,1,4),d(5,5,0,5,3,5,0,5,5,5,3,5),d(1,5,2,5,1,5,4,5,3,5,a,5),d(4,5,0,5,2,5,0,5,0,5)))
o(97,635,d(d(2,4,4,5,4,5,5,5,5,5,1,3),d(5,5,0,5,3,5,a,2,5,5,3,5),d(1,5,2,4,1,5,3,4,2,4,e,4),d(4,4,0,5,2,5,0,5,0,4)))
o(97,637,d(d(2,4,4,3,4,5,5,5,5,5,5,5),d(5,5,0,5,3,2,a,3,4,3,3,5),d(1,5,4,3,1,4,2,3,2,5,e,5),d(5,4,0,5,4,3,0,5,5,3)))
o(97,639,d(d(2,5,3,2,4,5,5,3,5,5,5,3),d(5,3,0,5,0,5,0,4,3,3,0,3),d(1,3,4,3,4,2,2,3,2,3,f,3),d(3,3,0,5,4,3,0,5,5,3)))
o(99,631,d(d(4,3,4,5,4,5,5,5,5,5,1,5),d(1,3,0,5,3,5,0,5,5,5,3,5),d(0,4,2,4,1,5,4,5,3,5,a,3),d(4,5,0,5,2,5,0,5,0,5)))
o(99,633,d(d(4,3,4,5,4,5,5,5,5,5,1,4),d(1,2,0,5,3,5,0,5,5,5,3,5),d(1,5,2,5,1,5,4,4,3,2,a,5),d(4,5,0,5,2,5,0,5,0,5)))
o(99,635,d(d(5,5,4,5,4,5,5,5,5,5,3,2),d(5,5,0,5,3,5,0,5,5,5,3,5),d(1,5,2,5,1,5,4,3,2,4,a,4),d(4,5,0,5,2,5,0,5,0,4)))
o(99,637,d(d(3,4,4,5,4,5,5,5,5,5,3,3),d(5,5,0,5,3,1,a,1,4,3,3,5),d(1,5,4,4,1,4,3,4,2,5,f,3),d(5,3,0,5,3,4,0,5,4,5)))
o(99,639,d(d(3,3,4,5,4,5,5,5,5,5,5,5),d(5,5,0,5,0,5,a,1,3,5,3,5),d(1,5,4,5,4,4,2,4,2,5,f,4),d(3,4,0,5,4,3,0,5,4,5)))