

```

1      a)
:- pred primeiro(list(T), list(T), list(T)).
:- mode primeiro(in, in, out) is det.
primeiro([],L2,L2).
primeiro([H|T],L2,[H|T2])
:- primeiro(T,L2,T2).

```

É um predicado determinístico, sucede sempre e tem uma solução única.
Recebe 2 listas e devolve a concatenação das duas.

```

      b)
:- pred segundo(list(T)::in,list(T)::out) is multi.
segundo([],[]).
segundo(L1,L2)
:- if delete(L1,X,R1) then segundo(R1,R2), L2=[X|R2] else L2=[].

```

É um predicado multi, sucede sempre e pode ter uma ou mais soluções.
Recebe uma lista e devolve todas as permutações possíveis dos elementos da lista.
Expo:

		RESULTADOS
[1,2,3]		
X=1 -> R1=[2,3]	->	[1,2,3] [1,3,2]
X=2 -> R1=[1,3]	->	[2,1,3] [2,3,1]
X=3 -> R1=[1,2]	->	[3,1,2] [3,2,1]

```

      c)
:- pred terceiro(pred(T1,T2),list(T1),list(T2)).
:- mode terceiro(pred(in,out) is det, in, out) is det.
terceiro(_,[],[]).
terceiro(P,[H1|T1],[H2|T2])
:- P(H1,H2),
   terceiro(P,T1,T2).

```

É um predicado determinístico, sucede sempre e tem uma solução única.
Aplica um predicado P a todos os elementos de uma lista e devolve os resultados através de uma lista de saída.

```

2      a) :- type highscoreTable ---> highscoreTable(int, int, list({string,int,int,int,int}));
      b) :- func init(NR,NJ) = highscoreTable(NR,NJ,[]);
      c) bestScore( (_,_,[{P1,S1,D1,M1,A1},T]),P2,S2,D2,M2,A2) :-
          { if P1 = P2 then
              S1=S2,
              D1=D2,
              M1=M2,
              A1=A2
            else
              bestScore( (_,_,T) , P2,S2,D2,M2,A2) }.

```

Grupo II

```

1      a) :- type condicoes ---> peixe_em_frente ; peixe_attingido ; deposito(int) ; arpoes(int).

```

:- type accoes ----> disparar ; agarrar ; reabastecer ; descarregar ; vaguear.

b) ----

2

KIF □ características gerais

□ O mundo é conceptualizado em termos de objectos e relações entre objectos

□ Uma relação é um conjunto arbitrário de listas de objectos.

□ Exemplo: a relação < é o conjunto de todos os pares (x,y) em que $x < y$.

□ O universo de discurso é o conjunto de todos os objectos cuja existência é conhecida, presumida ou suposta.

□ Os objectos podem ser concretos ou abstratos

□ Os objectos podem ser primitivos (não decomponíveis) ou compostos

KIF - meta-conhecimento

□ Pode formalizar-se conhecimento sobre o conhecimento

□ O mecanismo da citação (quotation) permite tratar expressões como objectos

□ Por exemplo a ocorrência da palavra □joão□ numa expressão designará uma pessoa; entretanto a expressão □(quote joão)□ ou □□joão□ designa a própria palavra □joão□ e não o objecto ou pessoa a que ela se refere.

□ Outros exemplos:

(acredita joão □(material lua queijo))

(=> (acredita joão ?p) (acredita ana ?p))

KIF é uma linguagem de comunicação entre Agentes

o KIF usa semantica puramente declarativa

tem termos e relações entre os termos...

agr,na pergunta kando diz onde s enquadra com a engenharia de conhecimento...

a eng. conhecimento é o processo ou actividade de construir bases de conhecimento

involve que se estude o domínio de aplicação, determinar os objectos e as relações, o vocabulario,

definir axiomas...

e é nisto k o KIF s enquadra

3

a) Pesquisa em profundidade sem repetição com limite crescente

limite 2

J - G - F

J - G - K

limite 3

J - G - F - C

J - G - F - B

J - G - K - E

J - G - K - I

limite 4

J - G - F - C - B

J - G - F - B - E

J - G - F - B - C

J - G - K - I - H

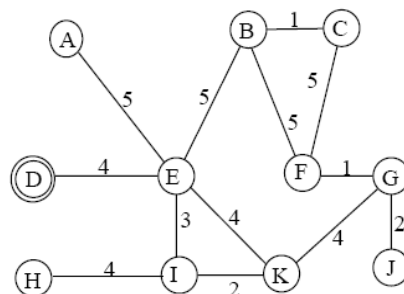
J - G - K - I - E

J - G - K - E - B

J - G - K - E - A

J - G - K - E - I

J - G - K - E - D



À parte(n faz parte do exercício mas convém saber -> pesquisa em largura)

J - G

J - G - F desdobramento do 1º nível

J - G - K

J - G - F - C desdobramento do 2º nível

J - G - F - B

J - G - K - I

J - G - K - E

J - G - F - C - B desdobramento do 3º nível

J - G - F - B - C

J - G - F - B - E

J - G - K - I - H

J - G - K - I - E

J - G - K - E - B

J - G - K - E - A

J - G - K - E - I

J - G - K - E - D

b) Ramificação Média = $(N-1) / X$

Factor de ramificação média -> $[B \cdot (B^{(d+1)}) - 1] / (B-1)$

N->nº nós da árvore no momento em q se encontra a solução

X->nº+ nós expandidos (não terminais)

d->comprimento(nº saltos) do caminho da árvore correspondente à solução

4. ???