Grupo VIII

1. Defina o predicado `tabuada(+N)` que dado um número inteiro `N`, apresenta no écran a tabuada do `N`.

2. Escreva um programa `escreve_tabuadas` que lê um inteiro do teclado, escreve no écran a sua tabuada, e continua pronto para escrever tabuadas até que seja mandado terminar.

3. Escreva um programa que lê uma lista de pares *(átomo, nº inteiro)* e apresenta um gráfico de barras dessa lista de pares. (Cada unidade deve ser representada pelo caracter #, e as barras podem ser horizontais.)

4. Escreva um programa que lê os coeficientes de um polinómio de 2º grau ax^2+bx+c e calcula as raizes reais do polinómio, apresentando-as no écran. Se o polinómio não tiver raizes reais, o programa deve informar o utilizador desse facto.

5. Relembre o problema apresentado anteriormente, em que a informação referente aos horários das salas de aula está guardada na base de conhecimento em factos da forma:

   `sala`(*num,dia,inicio,fim,discipl,tipo*)

   Defina um predicado `salva(+Ficheiro)` que guarda no `Ficheiro` os factos com a informação sobre as salas que são válidas (i.e., em que a hora de início é inferior à hora de fim).

6. Defina o predicado `findterm(+Term)` que escreve o écran o primeiro termo lido que unifica com `Term`.

7. Defina o predicado `findalltermsfile(+Term,+FileName)` que escreve no écran todos os termos do ficheiro que unificam com `Term` (garanta que `Term` não é instanciado).

8. Defina o predicado `to_upper(+FileIn,+FileOut)` que recebe um ficheiro de texto `FileIn` e gera o ficheiro `FileOut` com o mesmo texto de entrada mas convertido para letras maiúsculas. (Note que apenas as letras minúsculas são alteradas, o resto deverá ser mantido.)

9. Defina o predicado `numera_linhas(+FileIn,+FileOut)` que recebe o ficheiro `FileIn` e produz o ficheiro `FileOut`, que contém as mesmas linhas de `FileIn`, mas com as linhas numeradas.

10. Relembre o problema do cálculo da nota final à disciplina de *Lógica Computacional*, apresentado anteriormente, em que as notas do alunos estão guardadas na base de conhecimento em factos da forma:

    `modalidadeA`(*numero, nome, fichas, exame*)
    `modalidadeB`(*numero, nome, fichas, trabalho, exame*)

    Pretende-se agora poduzir a pauta final, tendo a informação sobre os alunos inscritos num ficheiro com factos da forma: `aluno`(*numero, nome, tipo*)

    Defina o predicado `gera_pauta(+Inscritos,+Pauta)` que dado o ficheiro `Inscritos`, vai lendo a informação sobre os alunos inscritos à disciplina e, sem alterar a base de conhecimento, produz no ficheiro `Pauta` o texto com a pauta devidamente preenchida. Note que:

    - se o aluno está inscrito mas não se submeteu a avaliação, a sua nota será Faltou;
    - se o aluno tem alguma das componentes de avaliação inferior a 9, ou a média final arredondada inferior a 10, a sua nota será Reprovado;
    - nos restantes casos, será o arredondamento da média pesada (se quiser, pode escrever também a nota por extenso).