

Prova Final de Algoritmos e Estruturas de Dados I

06/07/2005

Perguntas comuns e suas respostas:

- P: Tenho uma dúvida na questão tal.
R: A compreensão do enunciado faz parte da prova.
- P: Posso fazer a prova a lapis?
A prova é um documento, portanto deve ser feita à caneta.
- P: O que será corrigido?
R: Lógica, criatividade, sintaxe, uso correto dos comandos, a correta declaração dos tipos, os nomes das variáveis, endentação e clareza. Modularidade é importante, por isto use corretamente funções procedimentos, passagem de parâmetros, variáveis globais e locais.

Questão 1

(40 pontos)

Abaixo temos parte da implementação de um jogo de caça-palavras. Neste jogo de caça-palavras as palavras podem estar dispostas apenas na horizontal (da esquerda para direita apenas) ou na vertical (de cima para baixo). A entrada da matriz de letras deve ser feita a partir de um arquivo e as palavras devem ser entradas a partir do teclado. Para cada palavra a ser procurada devem ser impressa as coordenadas da matriz onde a palavra se inicia e onde termina. Você deve completar esta implementação para que o programa funcione corretamente.

```
Program Caca_palavras (input,output,arquivo);
```

```
(* declare as constantes necessárias *)  
(* declare os tipos necessários *)  
(* implemente os procedimentos e funções necessários *)  
(* declare as variáveis necessárias *)
```

```
BEGIN
```

```
    ASSIGN (arquivo,'matriz.txt');  
    RESET (arquivo);  
    Leia (tabuleiro, numLinhas, numColunas);  
    CLOSE (arquivo);  
    WRITELN ('entre com uma palavra qualquer');  
    READ (palavra);  
    WHILE NaoVazia (palavra) DO  
        BEGIN  
            IF Achou (palavra,tabuleiro,linIni,colIni,linFim,colFim) THEN  
                BEGIN  
                    WRITE ('A palavra ',palavra,' ocorre a partir da coordenada ');  
                    WRITELN (linIni,colIni,' terminando em ',linFim,colFim);  
                END;  
            READ (palavra);  
        END;
```

```
END.
```

Questão 2
(60 pontos)

Considere as seguintes definições de tipos:

```
TYPE
  coordenada = RECORD
                linha, coluna: integer;
            END;
  vetor = ARRAY [1..MAXTAM] of coordenada;
  ocorrencias = RECORD
                numOcorre: integer;
                coord    : vetor;
            END;
  tabela = ARRAY ['A'..'Z'] of ocorrencias;
```

A idéia consiste em tentar otimizar a busca de palavras armazenando as ocorrências de cada letra da matriz de entrada em uma tabela de ocorrências. Na tabela podemos registrar não apenas quantas vezes cada letra ocorre na matriz mas também as coordenadas de todas estas ocorrências.

1. (5 pontos) Faça um desenho da estrutura de dados “tabela” acima definida e explique como ela pode ser usada para que possamos saber, para cada letra, as coordenadas de todas as ocorrências desta letra na matriz do caça-palavras;
2. (30 pontos) Crie um procedimento que recebe como parâmetros a matriz do jogo, suas dimensões, e inicialize a estrutura da tabela conforme você acaba de explicar no item anterior;
3. (30 pontos) Implemente a função “Achou” do programa anterior de maneira a tirar onde ela ocorre, que podem ser armazenadas em um vetor. proveito da estrutura “tabela” aqui definida.