

Programação para Computação

Prova 2 – 26/11/2008

Prof. Marcus Vinícius Midena Ramos

Engenharia de Computação – UNIVASF

Questão 1 (3 pontos):

Manipulação de agregados: Construa uma função F que aceite como entrada uma matriz M de 5 linhas e 5 colunas, cujos elementos são do tipo inteiro, e gere como saída um vetor N com 5 elementos do tipo:

```
struct elem {
    int menor;
    int maior;
    int col_menor;
    int col_maior;
}
```

A função F deverá pesquisar as linhas da matriz M e, para cada linha analisada, atribuir na posição correspondente do vetor de saída N um elemento do tipo struct elem, tal que:

- O campo `menor` representa o menor valor encontrado na linha correspondente da matriz M;
- O campo `maior` representa o maior valor encontrado na linha correspondente da matriz M;
- O campo `col_menor` representa o número da coluna onde o menor valor foi encontrado na respectiva linha.
- O campo `col_maior` representa o número da coluna onde o maior valor foi encontrado na respectiva linha.

Caso haja números repetidos na mesma linha, o resultado poderá reportar qualquer um deles. Exemplo de entrada:

1	2	3	4	5
5	4	3	2	1
2	6	3	0	5
6	2	3	7	5
8	2	1	4	5

Saída correspondente (menor,col_menor # maior,col_maior):

```
1,0 # 5,4
1,4 # 5,0
0,3 # 6,1
2,1 # 7,3
1,2 # 8,0
```

O programa deverá exibir o conteúdo do vetor N na tela após o retorno da função F.

Questão 2 (3 pontos):

Numeração das linhas de um arquivo texto: Construa um programa que aceite três argumentos na linha de entrada, conforme descrito abaixo:

- O primeiro argumento corresponde ao nome de um arquivo de entrada a ser processado;
- O segundo argumento corresponde ao nome do arquivo de saída a ser gerado pelo programa; e
- O terceiro argumento corresponde a uma cadeia de caracteres que deverá ser pesquisada no arquivo de entrada.

O programa deve criar o arquivo de saída a partir do outro fornecido como entrada, de tal forma que o arquivo de saída reproduza o conteúdo do arquivo de entrada, com as seguintes modificações (supor que o arquivo de entrada contém cadeias de caracteres de comprimento máximo 80 e sempre terminadas pelo caracter '\0'):

- As linhas do arquivo de saída devem ser numeradas a partir de 1;
- O campo de numeração das linhas deve ocupar 5 posições e os números das linhas devem ser alinhados à direita;
- As linhas que são copiadas do arquivo de entrada devem iniciar na coluna 9 na linha correspondente no arquivo de saída;
- A coluna 7, no arquivo de saída, deve ser usada para indicar, através do caracter '*', se a correspondente linha contém a subcadeia fornecida como terceiro argumento na linha de comando; caso não contenha, ela deverá ser deixada em branco.
- As colunas 6 e 8 devem ser deixadas em branco.

Linha de comando:

```
converte entrada.txt saida.txt bcd
```

Exemplo de entrada (arquivo entrada.txt):

```
a
ab
abc
abcd
abcde
```

Exemplo de saída correspondente (arquivo saida.txt):

```
1  a
2  ab
3  abc
4 * abcd
5 * abcde
```

O programa deverá exibir o conteúdo do arquivo de saída na tela após a sua criação.

Questão 3 (4 pontos):

Batalha Naval Simplificada: construa um programa que crie uma representação de 10 posições, numeradas de 1 a 10, para um cenário de batalha naval. O único navio desse jogo ocupa 2 posições contíguas do total de 10 que compõem o cenário. A posição do navio é aleatória, sorteada pelo próprio programa no início da sua execução, e não é informada ao jogador.

O jogador pode fazer até 6 tentativas de tiros para afundar o navio. Para realizar um tiro, o jogador deverá informar um número de 1 a 10 ao programa. Para cada tiro efetuado pelo jogador, o programa deverá informar o mesmo se o navio foi atingido ou não. Caso o navio tenha sido atingido, essa informação deverá ficar registrada no programa.

O jogo é ganho se, em até 6 tentativas, o jogador conseguir atingir as duas posições do cenário onde o navio está posicionado. Caso contrário, ele perde o jogo. O jogo encerra apenas quando o jogador ganha ou perde, e pode terminar após 2, 3, 4, 5 ou 6 tentativas, dependendo dos resultados obtidos em cada uma delas.

Sugestão: use a função abaixo, que gera um número aleatório inteiro entre 1 e n, onde n é um parâmetro passado para a mesma:

```
int aleat (int n) {
    srand(time(NULL));
    return (rand()%n)+1;
}
```