

Programação para Computação

Prova Final – 04/12/2008

Prof. Marcus Vinícius Midena Ramos

Engenharia de Computação – UNIVASF

Questão 1 (1.5 pontos):

Escreva um algoritmo em Visualg que:

- Leia 9 valores inteiros do teclado e preencha uma matriz M de 3 linhas e 3 colunas com eles;
- Leia dois números inteiros a e b do teclado.

Na seqüência, o algoritmo deve verificar:

- Se o número a é maior ou igual a 1 e menor ou igual a 3;
- Se o número b é maior ou igual a 1 e menor ou igual a 3;
- Se o valor armazenado na linha a e na coluna b da matriz M contém um valor diferente de 0;

O algoritmo deverá encerrar a execução quando as três condições acima forem simultaneamente verificadas. Caso contrário, ele deverá fazer novas leituras de dados, até que essas condições sejam satisfeitas. Ao término, ele deverá imprimir na tela os valores de a e de b .

```
algoritmo "semnome"
var M: vetor [1..3,1..3] de inteiro
    i,j: inteiro
inicio
para i de 1 ate 3 faca
    para j de 1 ate 3 faca
        leia (M[i,j])
    fimpara
fimpara
repita
    repita
        escreva ("Digite a linha:")
        leia (i)
    ate (i>=1)e(i<=3)
    repita
        escreva ("Digite a coluna:")
        leia (j)
    ate (j>=1)e(j<=3)
ate M[i,j]<>0
escreva (i,j)
fimalgoritmo
```

Questão 2 (1.5 pontos):

Escreva um algoritmo em Visualg que leia 9 valores inteiros do teclado e preencha uma matriz M de 3 linhas e 3 colunas com eles. Na seqüência, o algoritmo deverá verificar se existe alguma linha, coluna ou diagonal principal preenchida apenas com zeros. O algoritmo deverá informar os números das linhas, das colunas e das diagonais principais para as quais essa condição seja verificada. Nota: existem duas diagonais principais, uma formada pelas posições [1,1][2,2][3,3] e outra formada pelas posições [3,1][2,2][1,3].

```
algoritmo "semnome"
var M: vetor [1..3,1..3] de inteiro
    i,j: inteiro
inicio
para i de 1 ate 3 faca
    para j de 1 ate 3 faca
        leia (M[i,j])
    fimpara
fimpara
se (M[1,1]=0) e (M[1,2]=0) e (M[1,3]=0) entao
```

```

    escreval ("Linha 1")
fimse
se (M[2,1]=0) e (M[2,2]=0) e (M[2,3]=0) entao
    escreval ("Linha 2")
fimse
se (M[3,1]=0) e (M[3,2]=0) e (M[3,3]=0) entao
    escreval ("Linha 3")
fimse
se (M[1,1]=0) e (M[2,1]=0) e (M[3,1]=0) entao
    escreval ("Coluna 1")
fimse
se (M[1,2]=0) e (M[2,2]=0) e (M[3,2]=0) entao
    escreval ("Coluna 2")
fimse
se (M[1,3]=0) e (M[2,3]=0) e (M[3,3]=0) entao
    escreval ("Coluna 3")
fimse
se (M[1,1]=0) e (M[2,2]=0) e (M[3,3]=0) entao
    escreval ("Diagonal 1")
fimse
se (M[3,1]=0) e (M[2,2]=0) e (M[1,3]=0) entao
    escreval ("Diagonal 2")
fimse
fimalgoritmo

```

Questão 3 (3 pontos):

Eliminação de espaços em branco. Construa uma função F (na linguagem C) com as seguintes características:

- F deve possuir dois parâmetros A e B do tipo string, com 80 caracteres cada;
- F deve gerar B a partir de A, da seguinte forma:
 - a. Todos os espaços em branco antes do primeiro caracter devem ser removidos;
 - b. Todos os espaços em branco que separam palavras consecutivas devem ser substituídos por um único espaço em branco;
 - c. Todos os espaços em branco do final do string devem ser removidos;
- Exemplos de mapeamento produzidos por F (as aspas são usadas apenas para realçar os strings e o símbolo □ é usado para representar espaços em branco):
 - "□□□abc□□def□ghij□□□□" produz "abc□def□ghij"
 - "ab□□□cde□□□f" produz "ab□cde□fg"
 - "□□abcd□efg□hi□□□□" produz "abcd□efg□hi"

```

#include <stdio.h>
void F (char A[], char B[]) {
    int i,j;
    i=0;
    j=0;
    while (A[i]!=' ') i++;
    while (A[i]!='\0') {
        while ((A[i]!=' ') && (A[i]!='\0')) {
            B[j]=A[i];
            i++;
            j++;
        }
        if (A[i]!=' ') {
            B[j]=A[i];
            i++;
            j++;
        }
        while (A[i]!=' ') i++;
    }
    B[j]=A[i];
    if (B[strlen(B)-1]==' ') B[strlen(B)-1]='\0';
}
int main () {
    char M[50];
    char N[50];
    printf ("Digite a cadeia de entrada:\n");
    gets (M);
}

```

```

F (M,N);
printf ("Resultado:\n");
printf ("\n%s\n",N);
}

```

Questão 4 (4 pontos):

Jogo da velha. Escreva um programa na linguagem C para gerenciar o jogo da velha entre dois jogadores. Crie uma representação para o cenário de um jogo da velha, com 3 linhas e 3 colunas, respectivamente numeradas 1 a 3 e 1 a 3. Deve-se gerenciar os lances de dois jogadores, respectivamente denotados por A e B. As jogadas de A e B deverão se alternar, até que um dos dois vença a partida. As jogadas deverão ser marcadas, no cenário, com a letra "A" para o jogador A e a letra "B" para o jogador B. A cada nova jogada o cenário completo e atualizado deverá ser mostrado na tela do computador.

Para jogar, o programa informar qual o jogador da vez (A ou B) e, em seguida, ler dois números inteiros correspondentes à linha e coluna desejadas por esse jogador. Ambos os números deverão variar de 1 a 3. Em caso de números inválidos ou de posições já ocupadas, novas leituras deverão ser feitas, até que apenas valores válidos sejam digitados. O primeiro a jogar é sempre o jogador A. Ganha o jogo o jogador que conseguir preencher uma mesma linha, uma mesma coluna ou uma diagonal do cenário com a mesma letra primeiro.

```

#include <stdio.h>
char V[3][3];

void iniciar () {
int i,j;
for (i=0;i<3;i++)
    for (j=0;j<3;j++) V[i][j]='.';
}

void jogar (char c) {
int linha, coluna;
do {
printf ("Vez do jogador %c\n",c);
do {
printf ("Digite a linha (1 a 3):\n");
scanf ("%d",&linha);
} while ((linha<1)|| (linha>3));
do {
printf ("Digite a coluna (1 a 3):\n");
scanf ("%d",&coluna);
} while ((coluna<1)|| (coluna>3));
} while (V[linha-1][coluna-1]!='.');
V[linha-1][coluna-1]=c;
}

int ganhou (char c) {
if (((V[0][0]==c) && (V[0][1]==c) && (V[0][2]==c)) ||
((V[1][0]==c) && (V[1][1]==c) && (V[1][2]==c)) ||
((V[2][0]==c) && (V[2][1]==c) && (V[2][2]==c)) ||
((V[0][0]==c) && (V[1][0]==c) && (V[2][0]==c)) ||
((V[0][1]==c) && (V[1][1]==c) && (V[2][1]==c)) ||
((V[0][2]==c) && (V[1][2]==c) && (V[2][2]==c)) ||
((V[0][0]==c) && (V[1][1]==c) && (V[2][2]==c)) ||
((V[2][0]==c) && (V[1][1]==c) && (V[0][2]==c))) return 1;
else return 0;
}

void mostrar () {
int i,j;
for (i=0;i<3;i++) {
for (j=0;j<3;j++) printf ("%c ",V[i][j]);
printf ("\n");
}
}

int main () {
iniciar ();
mostrar ();
while (1) {
jogar ('A');
mostrar ();
}
}

```

```
    if (ganhou ('A')) {
        printf ("Jogador A ganhou!\n");
        break;
    }
    jogar ('B');
    mostrar ();
    if (ganhou ('B')) {
        printf ("Jogador B ganhou!\n");
        break;
    }
}
}
```