

ÁREA II – CCEN / UFPE

Disciplina: **IF 165 – Computação Eletrônica** – 3º Exercício Escolar 2009.2 – **Gabarito** Data: 24 / 11 / 09

1º quesito: Tem-se N blocos paralelepípedos definidos por suas 3 arestas a, b e c e deseja-se determinar qual (quais) deste(s) bloco(s) tem (têm) face com maior área. Para isto pede-se fazer um programa em PASCAL que siga **rigorosamente** as instruções abaixo:

- a) Ler um inteiro N ($5 \leq N \leq 80$) correspondente ao número de blocos;
- b) Ler para os vetores BLA, BLB e BLC os valores das arestas a, b e c dos N blocos, utilizando um procedimento LERVET a ser incluído no programa; este procedimento deve ler um vetor cada vez que é chamado;
- c) Construir um vetor FC, contendo os valores da área da maior face de cada bloco, utilizando um procedimento CALCF. Este procedimento deve utilizar uma função MAIOR, que receba os valores das 3 arestas de um bloco e retorne com a área da maior face;
 - I. Obs: Para construir o vetor FC usando o procedimento CALCF, deve-se fazer uso da função MAIOR;
 - II. Obs: Passar o vetor FC como argumento em CALCF; passar as 3 arestas como argumento para MAIOR;
- d) Encontrar o valor da maior área entre todas as faces do vetor FC, utilizando uma função chamada MAIOR_FACE que recebe o vetor FC como argumento. (obs: caso tenha valores iguais como maior valor, retornar qualquer um dos valores).
- e) Imprimir, no programa principal, o valor da maior face encontrada em (d) e a relação dos blocos (número seqüencial que os define) que têm este valor como maior face.

Obs.: O programa principal deve conter os procedimentos e funções descritas abaixo, onde não deve ser utilizado variáveis globais:

- I. Um procedimento LERVET para ler um vetor de elementos reais;
- II. Um procedimento CALCF, para construir o vetor FC, que usará a função MAIOR (também incluída no programa) para verificar o valor da maior face;
- III. Uma função, MAIOR_FACE para encontrar a face de maior área entre todos os blocos paralelepípedo.

```
PROGRAM VOTO_SECRETO;
TYPE VET = ARRAY[1..80] OF REAL;
PROCEDURE LERVET( K : INTEGER; VAR V : VET );
VAR CONT: INTEGER;
BEGIN
    FOR CONT:= 1 TO K DO
        READLN( V[ CONT ] );
    END;
FUNCTION MAIOR( A, B, C: REAL ) : REAL;
VAR S1, S2, S3 : REAL;
BEGIN
    S1 := A * B; S2 := A * C; S3 := B * C;
    IF ( S1 >= S2 ) AND ( S1 >= S3 ) THEN MAIOR := S1;
    IF ( S2 >= S1 ) AND ( S2 >= S3 ) THEN MAIOR := S2;
    IF ( S3 >= S1 ) AND ( S3 >= S2 ) THEN MAIOR := S3;
END;
PROCEDURE CALCF( K: INTEGER; VA, VB, VC : VET; VAR VF : VET );
VAR CONT: INTEGER;
BEGIN
    FOR CONT := 1 TO K DO
        VF[ CONT ] := MAIOR( VA[ CONT ], VB[CONT], VC[CONT] );
    END;
FUNCTION MAIOR_FACE( K: INTEGER; VF: VET): REAL;
VAR CONT: INTEGER; MF: REAL;
BEGIN
    MF:= VF[ 1 ];
    FOR CONT:= 2 TO K DO
        IF VF[ ICONT ] > MF THEN MF:= VF[ CONT ];
    MAIOR_FACE := MF;
END;

VAR BLA, BLB, BLC, FC : VET;
    N, CONT : INTEGER;
    MF : REAL;
BEGIN
    REPEAT
        WRITE( 'Qual o número de blocos? '); READ( N );
    UNTIL ( N >= 5 ) AND ( N <= 80 );
    WRITE( ' Valores das faces A dos blocos ? '); LERVET( N, BLA );
    WRITE( ' Valores das faces B dos blocos ? '); LERVET( N, BLB );
    WRITE( ' Valores das faces C dos blocos ? '); LERVET( N, BLC );
    CALCF( N, BLA, BLB, BLC, FC );           { Construção do vetor FC }
    MF:= MAIOR_FACE( N, FC );           { Valor da maior face }
    WRITELN( 'Maior face = ', MF:7:2 );
    WRITELN( 'Número dos blocos com maior face = ', MF:7:2 );
    FOR CONT:= 1 TO N DO
        IF FC[CONT] = MF THEN WRITELN( ' Bloco ', I:3 );
    END.
```

Critério de correção

- | | |
|-------------------------------|-----|
| • Procedimento LERMAT | 1,0 |
| • Procedimento CALCF | 1,0 |
| • Função MAIOR | 1,0 |
| • Função MAIOR_FACE | 1,0 |
| • Programa principal | 1,0 |

2º quesito. Em uma eleição disputada por 3 candidatos de números 1,2 e 3, os votos dos eleitores foram armazenados no arquivo VOTOS.ARQ, contendo a seguinte estrutura de registros:

Título do Eleitor	Número do Candidato
String[15]	Inteiro

Obs.: os votos não válidos (nulos ou em branco) são armazenados com o número de candidato zero.

Fazer um programa em PASCAL para:

- Abrir o arquivo VOTOS.ARQ;
- Ler o arquivo VOTOS.ARQ e calcular o número de votos e a percentagem de votos válidos de cada candidato;
- Gravar o arquivo RES.ARQ com a quantidade de votos e percentagem de votos de cada candidato, utilizando a seguinte estrutura:

Número do Candidato	Total de Votos	Percentagem de Votos Válidos
Inteiro	Inteiro	Real

- Após a gravação do arquivo, imprimir na tela o conteúdo do arquivo RES.ARQ.

Obs.: a impressão deve ser feita a partir do arquivo RES.ARQ, utilizando os dados nele gravados e com os valores referentes a cada candidato impressos em linhas distintas.

PROGRAM CONTAGEM_DE_VOTOS;

```

TYPE REG1 = RECORD
    TE : STRING[15];
    NC: INTEGER;
END;
REG2 = RECORD
    NC, TV: INTEGER;
    PVV : REAL;
END;

VAR VOTOS: REG1;
    ARQ1 : FILE OF REG1;
    PERCV: REG2;
    ARQ2 : FILE OF REG2;
    NVC : ARRAY [ 1 .. 3 ] OF INTEGER; { Número de votos por candidate}
    I, TV: INTEGER;                   { Total de votos dos candidated }
    PVV : REAL                        { Percentagem de votos válidos}

```

Critério de correção

- | | |
|--|-----|
| • Definição das variáveis e ASSIGN's | 1,0 |
| • Abertura e leitura do arquivo VOTOS.ARQ .. | 1,0 |
| • Cálculo dos totais de votos e percentagens | 1,0 |
| • Gravação do arquivo RES.ARQ | 1,0 |
| • Saída dos resultados | 1,0 |

```

BEGIN
    ASSIGN( ARQ1, 'VOTOS.ARQ' );
    ASSIGN( ARQ2, 'RES.ARQ' );
    RESET(ARQ1);
    FOR I:= 1 TO 3 DO NVC[ I ]:= 0;
    WHILE NOT EOF(ARQ1) DO
    BEGIN
        READ( ARQ, VOTOS );
        IF VOTOS.NC = 1 THEN NVC[ 1 ]:= NVC[ 1 ] + 1;
        IF VOTOS.NC = 2 THEN NVC[ 2 ]:= NVC[ 2 ] + 1;
        IF VOTOS.NC = 3 THEN NVC[ 3 ]:= NVC[ 3 ] + 1;
    END;
    CLOSE( ARQ1 );
    REWRITE( ARQ2 );
    TV := NVC[ 1 ] + NVC[ 2 ] + NVC[ 3 ];
    FOR I:= 1 TO 3 DO
    BEGIN
        IF TV > 0 THEN PVV:= NVC[ I ] / TV * 100. ELSE PVV := 0;
        PERCV.NC := I;
        PERCV.TV := NVC[ I ];
        PERCV.PVV := PVV;
        WRITE( ARQ2, PERCV );
    END;
    CLOSE( ARQ2 );
    RESET( ARQ2 );

    WRITELN( ' Resultado da apuração ');
    WRITELN( ' Cand.  votos  perc.  ' )
    WHILE NOT EOF( ARQ2 ) DO
    BEGIN
        READ(ARQ2, PERCV );
        WRITELN(PERCV.NC:5; PERCV.TV:7, PERCV.PVV:8:3 );
    END;
    CLOSE( ARQ2 );
END.

```