**Final – 2011.2**

**1º quesito:** Escreva um programa em PASCAL que contenha as seguintes funcionalidades:

a) Leia um inteiro *n* que representa a dimensão de uma matriz quadrada de números reais. Validar o valor de *n* de modo que ele seja maior do que 1 e menor ou igual a 40.

b) Preencher (ler a partir do teclado) o conteúdo de uma matriz A por meio de um procedimento **Leitura** que receba como parâmetro a matriz A vazia, o seu tamanho *n* e retorne, para o programa principal, A preenchida.

c) Um procedimento **Transposta** que receba como parâmetros a matriz A, bem como seu tamanho *n*, realize o cálculo da transposta de A e armazene em uma matriz B --- a matriz B deve ser retornada ao programa principal.

d) Uma função **Máximo** que receba como parâmetro a matriz A e seu tamanho *n* e retorne, para o programa principal, o valor do maior elemento de A. Imprima na tela este valor.

e) Uma função **SomaTriangInferior** que receba como parâmetros a matriz B, o seu tamanho *n* e retorne para o programa principal o cálculo da soma de todos os elementos bij tal que i < j. Imprima na tela este valor.

f) Um procedimento **Imprime**, que receba como parâmetro uma matriz, seu tamanho e imprima na tela a matriz em formato matricial. Utilize esse procedimento para imprimir, respectivamente, as matrizes A e B.

**OBS.: Não deverão ser utilizadas variáveis globais.**

#include <stdio.h>

void leitura (float \*\*a, int n)

{

int i, j;

for (i=0; i<n; i++)

for (j=0; j<n; j++)

scanf (“%f”, &a[i][j];

}

void transposta (float \*\*a, int n, float \*\*b)

{

int i, j;

for (i=0; i<n; i++)

for (j=0; j<n; j++)

b[i][j] = a[j][i];

}

float máximo (float \*\*a, int n)

{

int i, j;

float maior = a[1][1];

if (a[i][j]>maior)

maior = a[i][j];

return maior;

}

float STriangInf (float \*\*b, int n)

{

int i, j;

float soma=0;

for (i=0; i<n; i++)

for (j=0; j<n; j++)

if (i<j)

soma += b[i][j];

return soma;

}

void imprimir (float \*\*x, int n)

{

int i, j;

for (i=0; i<n; i++)

{

for (j=0; j<n; ++)

printf (“%f ”, x[i][j]);

printf (“\n”);

}

}

int main ()

{

int n;

float a[40][40];

float b[40][40];

float m;

do

printf (“Digite o valor de n:”);

scanf (“%d”, &n);

} while (n<2 || n>40);

leitura (a, n);

transposta (a, n, b);

m = maximo(a, n);

printf (“\nO maior elemento da matriz A é %f\n”, m);

m = STriangInf(b, n);

printf (“A soma dos elementos da triangular inferior de A é %f\n”,m);

imprimir (a, n);

printf (“\n”);

imprimir (b, n);

return 0;

}

**2º quesito:** Uma empresa gostaria de oferecer um bônus de natal para seus funcionários, mas para isso quer relacionar o valor do bônus de natal com o salário e o número de dependentes de cada funcionário. Porém, estas informações estão dispostas em diferentes arquivos no setor de RH da empresa. Escreva um programa em PASCAL que leia os arquivos “Funcionarios.CAD”,

“Salarios.CAD” e “Cadastro.CAD”, relacione suas informações de forma a aplicar as seguintes regras para o cálculo do bônus de natal. Ao final do cálculo, salve no arquivo “Bonus.CAD” as informações calculadas. Abaixo são apresentadas as regras para o cálculo do bônus de natal e a estrutura dos arquivos.

**Regras:**

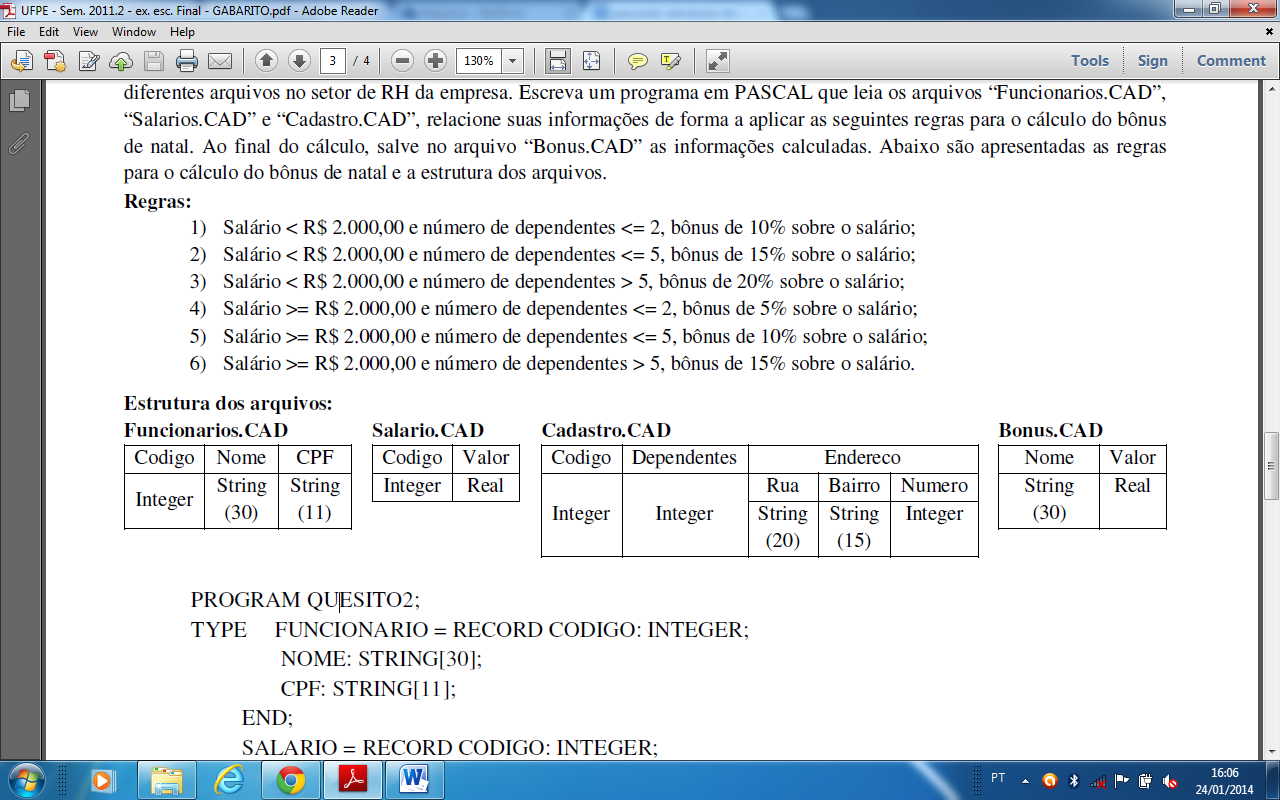
1) Salário < R$ 2.000,00 e número de dependentes <= 2, bônus de 10% sobre o salário;

2) Salário < R$ 2.000,00 e número de dependentes <= 5, bônus de 15% sobre o salário;

3) Salário < R$ 2.000,00 e número de dependentes > 5, bônus de 20% sobre o salário;

4) Salário >= R$ 2.000,00 e número de dependentes <= 2, bônus de 5% sobre o salário;

5) Salário >= R$ 2.000,00 e número de dependentes <= 5, bônus de 10% sobre o salário;

6) Salário >= R$ 2.000,00 e número de dependentes > 5, bônus de 15% sobre o salário.

#include <stdio.h>

struct funcionario

{

int codigo;

char nome [30];

char CPF[11];

};

struct salario

{

int codigo;

float valor;

};

struct endereco

{

char rua[20];

char bairro[10];

int numero;

};

struct cadastro

{

int codigo;

int dependentes;

char telefone[10];

endereco end;

};

struct bonus

{

char nome[30];

float valor;

};

int main ()

{

FILE \*arq1; FILE \*arq2; FILE \*arq3; FILE \*arq4;

funcionario func;

salario sal;

cadastro cad;

bonus bon;

float vbonus;

arq1 = fopen(“Funcionarios.CAD”, “r”);

arq2 = fopen(“Salarios.CAD”, “r”);

arq3 = fopen(“Cadastro.CAD”, “r”);

arq4 = fopen (“Bonus.CAD”, “w”);

while ((c=getc(arq1)) != EOF)

{

fread(func, sizeof(func), 1, arq1);

while ((c=getc(arq2)) != EOF)

{

fread(sal, sizeof(sal), 1, arq2);

if (func.codigo == sal.codigo)

{

while ((c=getc(arq3)) != EOF)

{

fread(cad, sizeof(cad), 1, arq3);

if (func.codigo == cad.codigo)

{

if (sal.valor<2000)

{

if (cad.dependentes<=2)

vbonus = sal.valor\*0.1;

else

if (cad.dependentes<=5)

vbonus = sal.valor\*0.15;

else

vbonus = sal.valor\*0.2;

}

else

{

If (cad.dependentes<=2)

vbonus = sal.valor\*0.05;

else

if (cad.dependentes<=5)

vbonus = sal.valor\*0.1;

else

vbonus = sal.valor\*0.15;

}

bonus.nome = func.nome;

bonus.valor = vbonus;

fwrite (bonus, sizeof(bonus), 1, arq4);

}

}

}

}

}

fclose(arq1);

fclose(arq2);

fclose(arq3);

fclose(arq4);

return 0;

}