Universidade Federal de Ouro Preto - UFOP
Disciplina: BCC 201 - Introdução à Programação
Professor: Guillermo Cámara-Chávez

Aluno: _____ **No.** _____

A cola não será tolerada. Se alguém for pego colando, será reprovado com Zero. É considerado cola: olhar/copiar da prova de outro ou deixar outro aluno olhar sua prova.

3ra. Avaliação - Grupo A

1. (2pts) Você deverá implementar uma estrutura TConjunto para representar conjuntos de números inteiros. Sua estrutura deverá armazenar os elementos do conjunto e o seu tamanho n. Considere que o tamanho máximo de um conjunto é 20 elementos e use arranjos de 1 dimensão (vetores) para a sua implementação. Crie a função que permite verificar se dois conjunto são iguais (os dados não se encontram necessáriamente ordenados).

```
typedef struct Conjunto{
    int elem[20];
    int n; // numero de elementos dentro
}TConjunto;

int Igual(TConjunto A, TConjunto B)
{
    int i, j;
    for (i = 0; i < A.n; i++)
    {
        for (j = 0; j < B.n; j++)
            if (A.elem[i] == B.elem[j])
                break;
        if (j == B.n)
            return 0;
    }
    return 1;
}

int main()
{
    TConjunto A = {{3, 5, 7, 8},5};
    TConjunto B = {{8, 7, 5, 3},4};
    if (Igual(A,B) == 1)
        printf("sao iguais");
    else
        printf("sao diferentes");
    return 0;
}
```

2. (2pts) Crie uma função que permita alocar memória para uma matriz tridimesional. O tamanho da matriz deve ser fornecido como parâmetro de entrada. Utilize a menor quantidade possível de malloc/calloc.

   Com número $lin + lin * col + 1$ calloc/malloc

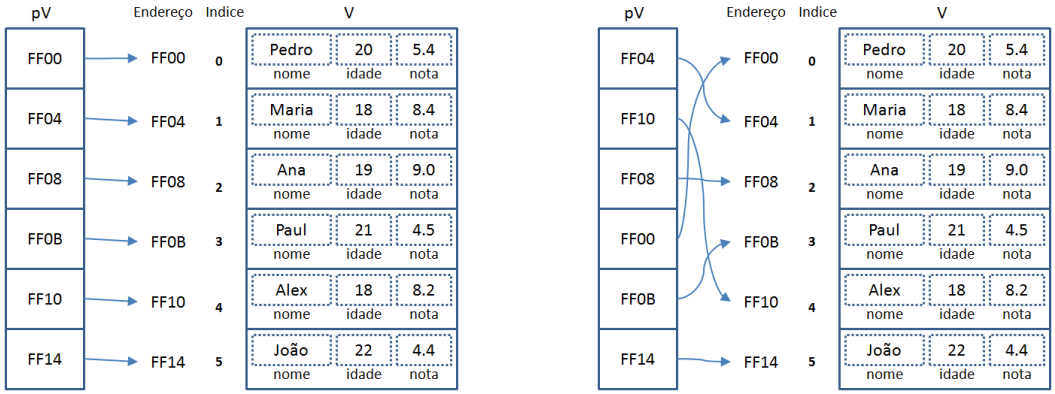
```c
int*** CriaMatriz3D(int lin, int col, int dim)
{
    int*** mat = NULL, i, j;
    mat = (int***) calloc (lin, sizeof(int**));
    for (i = 0; i < lin; i++){
        mat[i] = (int**) calloc (col, sizeof(int*));
        for (j = 0; j < col; j++)
            mat[i][j] = (int*) calloc (dim, sizeof(int));
        }
        return mat;
}
```

Com 3 calloc/malloc

```c
int*** CriaMatriz3Dmin(int lin, int col, int dim)
{
    int*** mat = NULL, i;
    mat = (int***) calloc (lin, sizeof(int**));
    mat[0] = (int**) calloc (lin*col, sizeof(int*));
    for (i = 1; i < lin; i++)
        mat[i] = &mat[0][i*col];
    mat[0][0] = (int*) calloc (lin*col*dim, sizeof(int));
    for (i = 1; i < lin*col; i++)
        mat[0][i] = &mat[0][0][dim*i];
    return mat;
}
```

3. (2pts) Faça uma função **MAX** que recebe como entrada um inteiro n, uma matriz inteira $A_{n \times n}$ (utilize ponteiros) e devolve um inteiro: k e dois vetores de inteiros l e c. k é o maior elemento de A e é igual a $A[l_i][c_i]$. Se o elemento máximo ocorrer mais de uma vez, retornar todas as posições onde se encontra o máximo elemento.

4. (2 pts) Elabore um procedimento/função que receba por parâmetro um vetor de estruturas TAluno de n posições e um vetor de ponteiros a estruturas TAluno de tamanho n. Faça que cada elemento do vetor de ponteiros contenha o endereço da respectiva posição do vetor de estruturas. Logo, ordene de forma ascendente o vetor de estruturas pelo campo idade, faça a ordenação a través do vetor de ponteiros. Para ordenar os dados, modifique somente as posições do vetor de ponteiros




```c
typedef struct Aluno{
    char nome[50];
    int idade;
    float nota;
}TAluno;

void OrdenaIdade(TAluno* V, TAluno** pV, int n){
```

```
    int i,j;
    TAluno *tmp;
    for (i = 0; i < n; i++)
        pV[i] = &V[i];
    for (i = 0; i < n; i++)
        for (j = 0; j < n-1; j++){
            if (pV[j]->idade > pV[j+1]->idade)
            {
                tmp = pV[j];
                pV[j] = pV[j+1];
                pV[j+1] = tmp;
            }
        }
}
```

5. (2pts) Considere um arquivo de dados do tipo texto com o seguinte conteúdo (lembre que é um exemplo!):

```
3
ZE SA
8.5
10.0
ANTONIO SANTOS
7.5
8.5
SEBASTIAO OLIVEIRA
5.0
6.0
```

O arquivo acima é um exemplo. Considere então que nestes arquivos a primeira linha contém o número de alunos no arquivo. As linhas seguintes contém os seguintes dados:

- nome do aluno com no m´aximo 50 caracteres;
- nota da primeira prova;
- nota da segunda prova.

Escreva um programa que imprima os nomes de todos os alunos que têm a média das duas notas menor que 7.0