**Nome: Eduardo Lucas Lemes Januário CT3037568**

**Nome: Guilherme Batista**

**Estrutura de Dados I - Igor Calebe Zadi**

**Lista de Exercícios 02**

1. Considere o exemplo do exercício 2), aqui repetido, queretrata a sintaxe necessária para alocar memória a fim de armazenar um arranjo unidimensional (vetor) V de inteiros de N posições pelo uso de um ponteiro.

Int \* V;

V = (int\*) calloc (N, sizeof(int));

1. Com base nessa sintaxe, como seria a alocação de memoria para um arranjo A que contivesse M ponteiros para inteiros?

Int \*\*A;

A = (int\* \*) calloc (M, sizeof (int\*))

1. Suponha que cada um dos ponteiros armazenados no arranjo A “aponte” para a sua própria região de N inteiros. Como seria a alocação de memória para a região “apontada” pelo ponteiro A[ M ]?

A[M] = (int \*) calloc (N, sizeof (int));

1. A situção apresentada no item b), equivale, na prática, a armazenar uma matriz bidimensional A em áreas não adjacentes. Indique como seria possivel usar a notação ponteiro-deslocamento para encontrar um elemento A[ i ] [ j ] dessa matriz. Admita que os indices dos arranjos tem valor zero na primeira posição.

\*(\*(A + i) + j)

1. Você acha que seria possível utilizar a notação ponteiro-índice neste caso? Justifique.

Sim, seria possível a notação ponteiro-índice, pois já ocorreu a organização dos ponteiros.

1. Suponha que você preicsa armazenar a lista de nomes dos seus colegas de classe. Com base no exercício 1) acima, como você faria isso utilizando arranjos alocados por ponteiros?

#include<stdio.h>

#include<stdlib.h>

#define Num\_alunos 3

#define Tam\_nome 50

Int main() {

Char\*\* nomes;

Nomes = (char\* \*) calloc (Num\_alunos, sizeof(char\*));

For (int i0; i<Num\_alunos; i++){

Nomes[ i ] = (char\*) calloc (Tam\_nome, sizeof(char));

}

Char alunos [Num\_alunos] [Tam\_nome] = {“Pedro”, “Lucas”, “Eduardo”}

For (int i=0; i<Num\_alunos; i++){

Nomes[ i ] [ 0 ] = alunos [ i ] [ 0 ]

For (int j = 1; alunos [ i ] [ j ] != “\0”; j++){

Nomes[ i ] [ j ] = alunos[ i ] [ j ];

}

}

Return 0;

}

1. Utilizando alocação dinâmica, escreva um programa que aloca uma matriz de 3 dimensões e preenche cada elemento dessa matriz com a soma dos índices do elemento. Por exemplo, Matriz[1] [2] [3] = 1 + 2 + 3 = 6.

#include<stdio.h>

#include<stdlib.h>

#define x 3

#define y 3

#define z 3

Int main() {

Int\*\*\* matriz;

Matriz = (int\*\* \*) malloc (x\* sizeof (int\*\*));

For (int i0; i<x; i++){

Matriz [ i ] = (int\* \*) malloc (y\* sizeof (int\*));

}

For (int i=0; i<x; i++){

For (int j =0; j<y; j++){

Matriz [ i ] [ j ] = (int\*) malloc (z\* sizeof(int));

}

}

//Preenche a matriz com a soma dos índices

For (int i=0; i<x; i++){

For (int j =0; j<y; j++){

For (int k=0; k<z; k++){

Matriz [ i ] [ j ] [ k ] = i + j + k;

}

}

}

//Exibir Valores

For (int i=0; i<x; i++){

For (int j =0; j<y; j++){

For (int k=0; k<z; k++){

printf(“Matriz [%d] [%d] [%d] = %d\n”, i, j, k,

Matriz[ i ] [ j ] [ k ]);

}

}

}

}