



UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Disciplina: **Programação Procedimental**
Curso: **Ciência da Computação**
Professora: **Elaine Ribeiro Faria**

Aula Prática no 13

Objetivo

Praticar o conceito de alocação dinâmica de memória usando o CodeBlocks.

1. Codifique, compile e execute um programa em C que:
 - a) Aloque dinamicamente um vetor formado por N números inteiros, sendo N um valor informado pelo usuário;
 - b) Peça ao usuário para digitar os N números no espaço alocado;
 - c) Imprima na tela os N números digitados;
 - d) Libere a memória alocada.

Não esqueça de usar funções para estruturar o seu código.

2. Codifique, compile e execute um programa em C que declare na função principal uma estrutura para o cadastro de alunos de uma academia.
 - a) Para cada aluno armazenar: nome, sobrenome (apenas um), ano de nascimento e um vetor de 6 posições indicando quais dias da semana o aluno frequenta a academia. Lembrando que a academia funciona de segunda a sábado;
 - b) Ao iniciar o programa, o usuário deverá informar o número de alunos que serão armazenados;
 - c) O programa deverá alocar dinamicamente a quantidade necessária de memória para armazenar os registros dos alunos;
 - d) O programa deverá pedir ao usuário que entre com as informações dos alunos;
 - e) Ao final, o programa deve exibir na tela os dados armazenados e liberar a memória alocada.

Não esqueça de usar funções para estruturar o seu código.

3. Codifique, compile e execute um programa em C que contenha uma ou mais estruturas para representar os dados de um funcionário de uma empresa. O tipo Funcionário deve possuir os seguintes campos:

- Nome do funcionário
- Data de nascimento
- Endereço completo
- Salário

O programa deve alocar dinamicamente um vetor de N posições, sendo cada posição ocupada por um funcionário da empresa. Lembrando que N deve ser um número fornecido pelo usuário do programa. Em seguida, crie 2 funções, uma função que preencha os dados do funcionário e outra função que imprima na tela os nomes dos funcionários que possuem o maior salário.

4. Codifique, compile e execute um programa em C que calcule a soma de duas matrizes $M \times N$ de

números reais (double). A implementação deste programa deve considerar as dimensões fornecida pelo usuário (Dica: represente a matriz através de variáveis do tipo double **, usando alocação dinâmica de memória). Não esqueça de criar uma função para liberar a área de memória alocada pela matriz.

5. Codifique, compile e execute um programa em C que contenha uma função que receba como parâmetros dois vetores de inteiros: x1 e x2 e as suas respectivas quantidades de elementos: n1 e n2. A função deverá retornar um ponteiro para um terceiro vetor, x3, alocado dinamicamente, contendo a concatenação de x1 e x2 e usar o ponteiro qtd para retornar o tamanho de x3. Sendo x1 = {1, 3, 5, 6, 7} e x2 = {1, 3, 4, 6, 8}, x3 irá conter {1, 3, 5, 6, 7, 1, 3, 4, 6, 8}. Assinatura da função:

```
int* concatena(int *x1, int *x2, int n1, int n2, int* qtd);
```

6. O que faz o código abaixo?

```
#include <stdio.h>

#define Y 4
#define X 4
int main(){
    int **A; int i,j;
    A = (int**)malloc(Y * sizeof(int*));
    for(i = 0; i < Y; i++)
        A[i] = (int*)malloc(X * sizeof(int));
    for(i = 0; i < Y; i++)
        for(j = 0; j < X; j++)
            A[i][j]=i+j;
    for(i = 0; i < Y; i++){
        for(j = 0; j < X; j++)
            printf("%d \t",A[i][j]);
        printf("\n");
    }
    for(i = 0; i < Y; i++)
        free(A[i]);
    free(A);
    return 0;
}
```