



LISTA DE EXERCÍCIOS – ALGORITMOS E ESTRUTURA DE DADOS III

Lista de Exercícios 2

DOCENTE: Thiago França Naves

DATA: __/__/__

ALUNO: _____

1) Sendo p um ponteiro, qual a diferença entre:

- p++;
- (*p)++;
- *(p++);
- O que quer dizer *(p+10);

2) Escreva um comentário em cada comando de atribuição explicando o que ele faz e o valor da variável à esquerda do '=' após sua execução.

```
int main()
{
    int y, *p, x;
    y = 0;
    p = &y;
    x = *p;
    x = 4;
    (*p)++;
    x--;
    (*p) += x;
    printf("y = %d\n", y);
    return(0);
}
```

3) Verifique o programa abaixo. Encontre o seu erro e corrija-o para que escreva o número 10 na tela.

```
#include <stdio.h>
int main() {
    int x, *p, **q;
    p = &x;
    q = &p;
    x = 10;
    printf("\n%d\n", &q);
    return(0);
}
```

- 4) Crie um vetor V com n inteiros, onde n é um valor inteiro fornecido pelo usuário. O vetor só deve ser alocado na memória depois que o usuário fornecer o valor de n.
- 5) Crie um programa para manipular vetores com ponteiros. O seu programa deve implementar uma função chamada `inverte_vetor`, que recebe como parâmetro dois vetores V1 e V2, ambos de tamanho N. A função deve copiar os elementos de V1 para V2 na ordem inversa. Ou seja, se a função receber $V1 = \{1,2,3,4,5\}$, a função deve copiar os elementos para V2 na seguinte ordem: $V2 = \{5,4,3,2,1\}$. Além disso, a função também deve retornar o maior valor encontrado em V1. A função deve obedecer ao seguinte protótipo: `int inverte_vetor(int *v1, int *v2, int n)`.
- 6) Escreva um programa em linguagem C que imprimam na tela a string s na ordem inversa utilizando ponteiros.
- ```
char s[10] = "abcde";
char* cptr;
// Seu código começa aqui
```
- 7) Escrever um programa que cadastre vários produtos em uma estrutura/registo. Em seguida, imprima uma lista com o código e nome da cada produto. Por último, consulte o preço de um produto através de seu código.
- 8) Faça um programa que leia um valor n e crie dinamicamente um vetor de n elementos e passe esse vetor para uma função que vai ler os elementos desse vetor. Depois, no programa principal, o vetor preenchido deve ser impresso. Além disso, antes de finalizar o programa, deve-se liberar a área de memória alocada.
- 9) Construa um programa (main) que aloque em tempo de execução (dinamicamente) uma matriz de ordem m x n (linha por coluna), usando 1+m chamadas a função `malloc`. Agora, aproveite este programa para construir uma função que recebendo os parametros m e n aloque uma matriz de ordem m x n e retorne um ponteiro para esta matriz alocada. Crie ainda uma função para liberar a área de memória alocada pela matriz. Finalmente, crie um novo programa (main) que teste/use as duas funções criadas acima.