



## LISTA DE EXERCÍCIOS

### UNIDADE 6 - PONTEIROS

1. Declare variáveis de diferentes tipos de dados e imprima os seus respectivos endereços na memória.
2. Leia o código abaixo, execute-o e explique o que está sendo impresso.

```
#include <stdio.h>

int main()
{
    int i = 1234;
    printf (" i = %d\n", i);
    printf ("%i = %p\n", &i);
    return 0;
}
```

3. Dado o código abaixo, explique a segunda e a terceira linha de código do método main() e conclua a implementação imprimindo o valor de i.

```
#include <stdio.h>

int main()
{
    int i = 10;
    int *pi = &i;
    *pi = 20;
    ...
}
```

4. Leia o código abaixo, execute-o e explique o último valor impresso.

```
#include <stdio.h>

int main()
{
    int i = 10;
    int j = 5;
```

```

printf ("valor de i = %d\n", i);
printf ("valor de j = %d\n", j);
printf ("endereço de &i = %p\n", &i);
printf ("endereço de &j = %p\n", &j);

int *pi = &i;

printf ("endereço de &pi = %p\n", &pi);
printf ("endereço apontado por *pi = %p\n", pi);

*pi = *pi + j;

printf ("valor de i = %d\n", i);

return 0;

}

```

5. Escreva um programa que declare um inteiro, um real e um char, e ponteiros para inteiro, real, e char. Associe as variáveis aos ponteiros (use &). Modifique os valores de cada variável usando os ponteiros. Imprima os valores das variáveis antes e após a modificação.

6. Analise o seguinte código:

```

a. int numeros[5] = {1, 2, 3, 4, 5};
b. int *p;
c. p = numeros;
d. printf("%d", p[0]);
e. p++;
f. printf("%d", *p);

```

Responda:

- Que valor será impresso na linha (d)?
- O que acontecerá na linha (e)?
- Que valor será impresso na linha (f)?

7. Quais as instruções para alocar um espaço dinamicamente para 50 inteiros?
8. Escreva um programa em C para **ler** um vetor de 5 itens do tipo inteiro usando **aritmética de ponteiros**. Após a leitura, imprima os valores.
9. Explique, passo a passo, o efeito do seguinte fragmento de código:
- ```

int *v;
v = malloc (10 * sizeof (int));

```
10. Crie um vetor de inteiros com tamanho definido pelo usuário e alocado na memória dinamicamente. Insira os dados para o vetor e depois recupere os valores e imprima na tela.
11. O que há de errado com o seguinte fragmento de código?

```
int *v;  
v = malloc (100 * sizeof (int));  
v[0] = 999;  
free (v+1);
```

12. Crie uma *struct* qualquer, aloque dinamicamente um espaço na memória para armazenar os valores nela.

13.