

## Lista de Exercícios de Ponteiros

Programação de Computadores I  
Sistemas de Informação Ano 2019 – UEMS  
Professora: Mercedes Gonzales Márquez

1. Explique o significado de cada ocorrência de \* no fragmento de código a seguir e indique qual a saída exibida na tela.

```
int *p, x=5;
*p *= 2**p;
printf("%d", x);
```

2. Analise o seguinte programa e sua saída.

```
#include "stdio.h"
void main(void){
    int x = 5 , y = 6;
    int *px,*py;
    px = &x;
    py=&y;
    if(px<py)
        printf("py-px = %u\n",py-px);
    else
        printf("px-py = %u\n",px-py);
    printf("px = %u, *px = %d, &px = %u\n",px,*px,&px);
    printf("py = %u, *py = %d, &py = %u\n",py,*py,&py);
    px++;
    printf("px = %u, *px = %d, &px = %u\n",px,*px,&px);
    py=px+3;
    printf("py = %u, *py = %d, &py = %u\n",py,*py,&py);
    printf("py-px = %u\n",py-px);
}
```

3. Teste o programa a seguir para um nome da lista e para outro que não esteja na lista.

```
#include "stdio.h"
#include "string.h"
#define MAX 5
void main(void){
    int i, existe=0;
    char nome[40], *list[MAX]={"Maria","Gesiel","Lucas","Poliana","Juliana"};
    printf("Digite seu nome : ");
    scanf ("%s", nome);
    for(i=0;i<MAX;i++)
        if(strcmp(list[i],nome) == 0)
            existe = 1;
    if(existe)
        printf("Voce esta na lista, tem permissao para entrar !");
    else
        printf("Vc nao faz parte da lista ! ");
}
```

4. Mostre em uma tabela todos os passos (teste de mesa) e identifique qual será a saída do programa para três diferentes pares de valores lidos (x e y).

```
void teste(int *px, int *py) {
    px = py;
    *py = (*py) * (*px);
    *px = *px + 2;
}
int main(void) {
    int x, y;
    scanf("%d",&x);
    scanf("%d",&y);
    teste(&x,&y);
    printf("x = %d, y = %d", x, y);
}
```

5. Escreva uma função `length(s)` que receba como parâmetro uma string `s` e retorne seu tamanho (equivalente a função `strlen(s)` da biblioteca `string.h`).
6. Escreva uma função `copy(s,t)` que receba como parâmetro duas strings e copie o conteúdo da string `t` na string `s` (equivalente a função `strcpy(s,t)` da biblioteca `string.h`).
7. Escreva uma função `compare(s,t)` que receba como parâmetro duas strings e compare `s` e `t`, retornando um valor negativo, zero ou positivo se `s` for, respectivamente, lexicograficamente menor, igual ou maior que `t` (equivalente a função `strcmp(s,t)` da biblioteca `string.h`).
8. Escreva uma função `concatenate(s,t)` que receba como parâmetro duas strings e concatene `t` em `s` (equivalente a função `strcat(s,t)` da biblioteca `string.h`).
9. Suponha que você precise de uma variável para armazenar `n` strings lidas do teclado. É melhor declará-la como uma matriz de caracteres ou como um vetor de ponteiros? Justifique sua resposta.