



**CENTRO UNIVERSITÁRIO DE JOÃO PESSOA  
PRÓ-REITORIA ACADÊMICA - PROAC**

**CIÊNCIA DA COMPUTAÇÃO / ADS**

**Componente Curricular:** Técnicas e Desenvolvimento de Algoritmos

**Professor:** Ricardo Dantas

**Semestre:**  
2023.2

**Período:**  
1º/2º

**Turma:**  
Manhã/Noite

**Data de Entrega:**  
Ver Blackboard

## Lista de Exercícios – Ponteiros

Obs. Tentem responder sem reproduzir os códigos na IDE.

### 1. Qual é saída do programa em C abaixo?

```
1 int main(void){
2     int i = 99, j;
3     int *p;
4
5     p = &i;
6     j = *p + 100;
7     printf("i = %d, j = %d, *p = %d", i, j,*p);
8 }
```

### 2. Qual é saída do programa em C abaixo?

```
1 int main() {
2     int a = 5, b = 12, c;
3     int *p;
4     int *q;
5
6     p = &a;
7     q = &b;
8
9     c = *p + *q;
10
11     printf("c = %d, a = %d, b = %d", c, a, b);
12 }
```

3. Qual é saída do programa em C abaixo para os valores lidos (x = 3 e y = 4)?

```
1 void func(int * px, int * py){
2     px = py;
3     *py = (*py) * (*px);
4     *px = *px + 2;
5 }
6
7 int main() {
8     int x, y;
9
10    scanf("%d",&x); // 3
11    scanf("%d",&y); // 4
12
13    func(&x, &y);
14    printf("x = %d, y = %d", x, y);
15 }
```

4. Qual é saída do programa em C abaixo?

```
1 int main() {
2     int a = 4, b = 3, *p1, *p2;
3
4     p1 = &a;
5     p2 = p1;
6     *p2 = *p1 + 3;
7     b = b * (*p1);
8     (*p2)++;
9     p1 = &b;
10
11    printf("%d %d\n", *p1, *p2);
12    printf("%d %d\n", a, b);
13 }
```

5. A memória do computador funciona, basicamente, armazenando sequências de bytes, e cada sequência tem seu próprio tipo e endereço de memória. A quantidade de bytes de uma sequência é determinada pelo tipo, que pode ser: tipo primitivo, ponteiro, estrutura de dados ou a combinação ilimitada destes. O endereço de memória é representado por um número hexadecimal na forma 0x1234 (os valores apresentados nesta avaliação são imaginários e não representam endereços reais válidos).

Com base nestas informações, avalie as afirmações a seguir.

- I. Um ponteiro é um tipo de variável que pode ser manipulado com os operadores "&" e "\*".
- II. O que um ponteiro carrega dentro dele é um endereço de memória de uma sequência de bytes de um determinado tipo.
- III. É possível realizar operações aritméticas (soma e subtração) sobre ponteiros.

É correto o que se afirma em

- A) I, apenas.
- B) III, apenas.
- C) I e II, apenas.
- D) II e III, apenas.
- E) I, II e III.

6. A memória do computador funciona, basicamente, armazenando sequências de bytes, e cada sequência tem seu próprio tipo e endereço de memória. A quantidade de bytes de uma sequência é determinada pelo tipo, que pode ser: tipo primitivo, ponteiro, estrutura de dados ou a combinação ilimitada destes. O endereço de memória é representado por um número hexadecimal na forma 0x1234.

Com base nestas informações, avalie as afirmações a seguir.

- I. Para obter o endereço de memória de uma variável qualquer, utiliza-se o operador "&" antes do nome da variável.
- II. O que um ponteiro carrega dentro dele é um endereço de memória de uma sequência de bytes de um determinado tipo.
- III. Para obter os bytes que estão no endereço de memória que está dentro de uma variável do tipo ponteiro, utiliza-se o operador "\*" antes do nome da variável.

- IV. Para obter o conteúdo de uma parte de uma estrutura de dados, referenciada por uma variável do tipo desta estrutura, utiliza-se o operador "." (ponto) depois do nome da variável, seguido do nome da variável da parte desejada.

É correto apenas o que se afirma em

- A) I, II e IV.
- B) I, II, III e IV.
- C) II e III.
- D) I, III e IV.
- E) I e IV.