



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO – CAMPUS PAU DOS FERROS
BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO
DISCIPLINA: LABORATÓRIO DE ALGORITMOS E ESTRUTURAS DE DADOS I

LISTA DE EXERCÍCIOS - PONTEIROS

Questão 1) Quais serão os valores de x , y e p ao final do trecho de código abaixo:

```
int x, y, *p;  
y = 0;  
p = &y;  
x = *p;  
x = 4;  
(*p)++;  
--x;  
(*p) += x;
```

$y = 4.$
 $x = 3.$
 $p = \text{endereço de } y.$

Questão 2) Considere o trecho de código abaixo:

```
int main(void) {  
    int x, *p;  
    x = 100;  
    p = x;  
    printf("Valor de p = %p\tValor de *p = %d", p, *p);  
}
```

Se tentarmos compilar o programa (não o compile ainda), você acha que o compilador nos fornece alguma mensagem? Se sim, responda:

- a) Esta mensagem é de erro ou advertência? **Erro.**
- b) Por que o compilador emite tal mensagem? **Pois está tentando se atribuir um valor inteiro a um ponteiro.**
- c) Compile e execute o programa. A execução foi bem sucedida? **Não.**
- d) Modifique o trecho de código acima, de modo que nenhuma mensagem seja emitida pelo compilador. **A notação correta seria $p = \&x$;**
- e) Compile e execute novamente o programa. A execução foi bem sucedida?
Sim, dessa vez ele executou bem.

Questão 3) Qual será a saída exibida pelo programa a seguir:

```
#include <stdio.h>
int main(void) {
    int a, b, c, d;
    int *p1;
    int *p2 = &a;
    int *p3 = &c;
    p1 = p2;
    *p2 = 10;
    b = 20;
    int **pp;
    pp = &p1;
    *p3 = **pp;
    int *p4 = &d;
    *p4 = b + (*p1)++;
    printf("%d\t%d\t%d\t%d\n", a, b, c, d);
    return 0;
}
```

a = 10.
b = 30.
c = 10.
d = 30.

Observe que, se **int *p** é uma variável do tipo ponteiro para inteiro, então **int **pp** é uma variável do tipo ponteiro para ponteiro de inteiro, isto é, uma variável que poderá armazenar o endereço de um ponteiro do tipo int.

Questão 4) Crie um programa para calcular a área e o perímetro de um hexágono. O seu programa deve implementar uma função chamada `calcula_hexagono` que calcula a área e o perímetro de um hexágono regular de lado l . A função deve obedecer ao seguinte protótipo: `void calcula_hexagono(float l, float *area, float *perimetro);` A área e o perímetro de um hexágono regular são dados, respectivamente, por:

$$A = \frac{3l^2\sqrt{3}}{2} \quad P = 6l$$

Obs: Para os cálculos, utilize as funções `sqrt` e `pow` do módulo `math` da biblioteca padrão de C. Em seguida, crie a função principal do programa e utilize a função `calcula_hexagono` para calcular a área e o perímetro de um hexágono de lado l informado pelo usuário.