

Curso: Licenciatura em Computação

Disciplina: Estruturas de Dados

Professor: Edson Angoti Júnior

Instruções para entrega:

- 1. Postar os códigos dos exercícios no Github**
- 2. Entregar a tarefa no Google Classroom indicando o link do repositório Github**

Lista de exercícios sobre ponteiros

Aluno: Danilo dos Passos Terra

1) Explique a diferença entre:

`p++`; `P++`, faz a soma depois de usar a variável "p". ou seja `p=p+1`

`(*p)++`; `(*p)++`, faz `*p` e depois a soma de +1; sendo que `p++` é igual a `p+1`

`*(p++)`; `*(p++)`, faz `p+1` e depois a multiplicação.

- 2) O que quer dizer `*(p+10)`; ? O inteiro contido no endereço `p+10`, isto é, o décimo inteiro que se segue ao inteiro apontado por `p`.
- 3) Qual o valor de `y` no final do programa? Tente primeiro descobrir e depois verifique no computador o resultado. A seguir, 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; /* atribui o valor 0 a y => y=0 */
    p = &y; /* atribui o endereço de y ao ponteiro p
                p contem o endereço de y (ex:DS:FFF4)*/
    x = *p; ;          /* atribui o conteúdo de onde p aponta
                (valor de y) para x, que passa a valer 0 */

    x = 4; /* atribui 4 a x */
    (*p)++; /* incrementa de 1 o conteúdo de onde p aponta,
                alterando o valor de y para 1 */
    x--; /* decrementa 1 de x => x = 3 */

    (*p) += x; /* adiciona x ao conteúdo de onde p aponta,
                alterando o valor de y para 4 */

    printf ("y = %d\n", y); ("y = %d\n", y); /* imprime "y = 4" */
```

```
    return(0);  
}
```

- 4) Verifique o programa abaixo. Encontre o seu erro e corrija-o para que escreva o número 10 na tela (mantenha a variável *q* como parâmetro do *printf*).

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

O programa contém um erro na linha do *printf*, onde ele manda imprimir o endereço de **q** (&q). Na realidade, para se imprimir o valor 10 (valor de *x*) deve-se imprimir o valor apontado pelo valor apontado por **q**. Veja o esquema:

x = 10;

p aponta para *x*;

q aponta para **p**;

==> *q é igual a p ; como *p é igual a x, basta escrever *(*q) para se ter x. Logo, o printf ficaria:

```
printf("\n%d\n", **q);
```

5) Escreva um programa que declare uma matriz 100x100 de inteiros. Você deve inicializar a matriz com zeros usando ponteiros. Preencha depois a matriz com os números de 1 a 10.000 usando ponteiros.

```
/* Problema das matrizes ----- */
#include <stdio.h>
#define N 100

main ()
{
int mat[N][N];
int *p;
int i, j, soma = 0;
```

```
p = &mat[0][0];    /* Inicializa o ponteiro no inicio da matriz */

/* Inicializando a matriz com zeros.. */
for (i=0; i<N; i++)
    for (j=0; j<N; j++)
        {
            *p = 0;
            p++;
        }

/* Preenchendo a matriz com numeros */
p = &mat[0][0];
for (i=0; i<N; i++)
    for (j=0; j<N; j++)
        {
            *p = soma;
            soma++;
            p++;
        }
}
```

6) O valor de uma variável ou expressão do tipo vetor é o endereço do elemento zero do vetor.

Seja **a[]** um vetor qualquer, independente de tipo e tamanho, e **pa** um ponteiro para o mesmo tipo de **a[]**. Responda V ou F, justificando:

(V) Após a atribuição **pa=&a[0]**; **pa** e **a** possuem valores idênticos, isto é, apontam para o mesmo endereço

(V) A atribuição **pa=&a[0]**; pode ser escrita como **pa=a**; (V) **a[i]** pode ser escrito como ***(a+i)**

(V) **&a[i]** e **a+i** são idênticos

(V) **a+i** e' o endereço do i-ésimo elemento após **a**

(V) **pa[i]** e' idêntico a ***(pa+i)**

(V) **pa=a** e' uma operação valida (V) **pa++** e'

uma operação valida (F) **a=pa** e' uma operação

valida (F) **a++** e' uma operação valida

7) O que está errado com os programas abaixo? Descubra e indique a solução para consertá-los. Execute-o no computador para ver se o erro foi resolvido.

a)

```
void main(void) /* esse programa esta errado */ { int x,
    *p;
    x = 10;
    *p = x;
}
```

b) `void main(void) /* esse programa esta errado */ { int x, *p;
 x = 10;
 p = x;
 printf ("%d", *p);
}`

8) 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;
```