**Aluno: Samuel Figueiredo Silveira**

**1- Qual o valor de y no final do programa?**

#include <stdio.h>

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, que contém o endereço de y

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); (imprime "y = 4")

O valor de **y** é 4

**2-Teste e explique a diferença, caso exista, entre**

**p++;**

incrementa o ponteiro, ou seja, o endereço. Após esta instrução, o ponteiro p passará a apontar para a posição de memória imediatamente superior.

**(\*p)++;**

Incrementa o valor apontado por p, seu conteúdo.

**\*(++p);**

Incrementa **p** (como em p++) e acessa o valor encontrado na nova posição.

**b) O que quer dizer \*(p+10)?**

Acessa o valor encontrado 10 posições a frente de **p**. Neste caso, o apontador não é incrementado

**3-Quais serão os valores de x, y e p ao final do trecho de código**  
**abaixo?**  
int x, y, \*p;

y = 0;  
p = &y; (\*p = 0)  
x = \*p; (x = 0)  
x = 4;  
(\*p)++; (\*p = 1, y = 1)  
--x; (x=3)  
(\*p) += x; (\*p = 4, y = 4)

Ao final, temos: x = 3, y = 4, p apontando para y (\*p = 4)

**4-Os programas (trechos de código) abaixo possuem erros. Qual(is)?**

**Ponteiros\_gabarito.pdf**  
**Como deveriam ser?**  
**a)**  
**void main() {**  
**int x, \*p;**  
**x = 100;**  
**p = &x;**  
**printf(“Valor de p: %d.\n”, \*p);}**

p deveria receber o endereço de x, já que p é um ponteiro (e x não). Ponteiros “armazenam” o endereço para o qual eles apontam

**b)**  
**void troca (int \*i, int \*j) {**  
**int \*temp;**  
**\*temp = \*i;**  
**\*i = \* j;**   
**\*j = \*temp;**

**}**

void troca (int \*i, int \*j) {

int temp;

temp = \*i;

\*i = \*j;

\*j = temp;

}

A variável “temp” não precisava ser um ponteiro, já que apenas precisa armazenar um valor inteiro, sem precisar apontar para algum lugar

**c)  
char \*a, \*b;  
a = "abacate";  
b = "uva”;  
if (a < b)  
printf ("%s vem antes de %s no dicionário", a, b);  
else  
printf ("%s vem depois de %s no dicionário", a, b);**

char a[] = "abacate", b[] = "uva";

if (a[0] < b[0])

printf ("%s vem antes de %s no dicionário", a, b);

else

printf ("%s vem depois de %s no dicionário", a, b);

**Nesse caso, verificar apenas a primeira letra das cadeias de caracteres funciona, pois temos “a” e “u”**

**8- Escreva uma função chamada troca que troca os valores dos**  
**parâmetros recebidos. Sua assinatura deve ser:void troca(float \*a, float \*b);**

void troca (float \*a, float \*b) {

float temp;

temp = \*a;

\*a = \*b;

\*b = temp;

}