

### Prática 02 – Ponteiros

1. Observe o programa abaixo e identifique a saída do printf. Desenhe um esquema de memória contendo x, p1 e p2 (estime os endereços atribuídos a cada variável).

```
# include <stdio.h>

int main ()
{
int x, *p1, *p2;

p1 = &x;
p2 = p1;
printf ("%d", p2);
}
```

2. Execute o programa abaixo passo a passo no compilador e registre em papel os endereços alocados para cada variável (inclusive os ponteiros). Observe o que acontece no decorrer do programa e no final. Tente entender o que o programa faz e explique porque acontece o erro de execução ao final.

```
# include <stdio.h>

int main ()
{
int x, *p, *q, i, k=0;

q = p = &x;
for (i=1; i <= 3; i++)
{
scanf ("%d", p);
p++;
}
p--;
while (p >= q)
{
k = k + *p;
p--;
}
printf ("%d", k);
}
```

Inverta a ordem de declaração das variáveis e observe o que acontece.

3. Observe os programas abaixo e identifique o erro cometido em cada um no uso de ponteiros.

```
int main ()
{
    int x, *p;
    x = 10;
    *p = x;
}
```

---

```
int main ()
{
    int x, *p;
    x = 10;
    p = x;
    printf ("%d", *p);
}
```

---

```
int main ()
{
    char s[80], y[80];
    char *p1, *p2;
    p1 = s;
    p2 = y;
    if (p1 < p2) ....
}
```

---

```
int main ()
{
    int primeiro[10], segundo[10];
    int *p, t;
    p = primeiro;
    for (t = 0; t < 20; ++t)
        *p++ = t;
}
```

---

```
int main () /* deve ler várias strings, (até "fim") e mostrar cada string com
1 letra por linha */
{
    char *p1, s[80];
    p1 = s;
    gets (s);
    while (strcmp (s, "fim") != 0)
```

```

{
    while (*p1 != NULL)
        printf ("%c\n", *p1++);
    gets (s);
}
}

```

4. O que está errado com os programas abaixo? Descubra e indique a solução para consertá-los.

<pre> int main() { int x, *p; x = 10; *p = x; } </pre>	<pre> int main() { int x, *p; x = 10; p = x; printf ("%d", *p); } </pre>	<pre> //Este prog. deve mostrar 10 na tela #include &lt;stdio.h&gt; void int main() { int x, *p, *q; p = &amp;x; q = p; x = 10; printf ("\n%d\n", &amp;q); } </pre>
--	--	---

5. Para o código abaixo:

```

#include<stdio.h>
#include <alloc.h>
int main () {
int y, x, k, *p,*q;
p = &x;
q = (int *)malloc(6 * sizeof(int));
for(y = 0; y< 3; y=y+1) {
p--;
scanf("%d", p);
p++;
scanf("%d", p);
*q = k;
*(q+1) = x;
q = q + 2;
}
}

```

- Desenhe o esquema de memória mostrando a localização das variáveis do programa, até a linha 5 (inclusive). Mostre o conteúdo de p e q. Considere 100 como o endereço de início de atribuição das variáveis e 200 para a área de alocação de memória, o tamanho de inteiro como 4 bytes e o tamanho de um ponteiro como 4 bytes.

- b. Desenhe o esquema de memória mostrando a localização e conteúdo das variáveis do programa ao final da execução, se forem digitados: 50, 30, 20, 10, 40, 80.
- c. Indique o valor dos itens abaixo após a execução da linha 13. pela primeira vez:
  - i. `q[0]`
  - ii. `&q`
  - iii. `*(&k + 1)`