

PROGRAMAÇÃO II – SISTEMAS DE INFORMAÇÃO

Lista de exercícios 10 – Ponteiros – GABARITO

Prof. Jean Eduardo Glazar

- 1) Analise as seguintes sequências de código. Verifique se estão corretas ou incorretas. Caso estejam incorretas, apontem os prováveis erros. Para tanto, considere a declaração das variáveis abaixo:

`int a, b, *c, *d;`

a) `c = &a;`

b) `c = &b;` **correto**
`d = c;`
`*d = 10;`

c) `c = &a;`
`*d = 10 + *c;`

Não se sabe para onde "d" está apontando. Pode ter lixo.

d) **Não se sabe para onde "c" está apontando.**

`d = c;`
`c = &a;`
`a = 10;`
`b = *d;` **"d" pode ter lixo**
`printf("%d", b);`

e) `c = &a;`
`scanf("%d", c);`
`d = c;`
`b = *d;`

f) `b = 10;`
`c = &b;`
`printf("%d", *c);`

g) `*c = 20;` **Correto. Porém não se sabe para onde "c" está apontando.**

h) `b = 2;` **Correto**
`c = &b;`
`*c += 2;`

i) `b = 1;`
`c = &b;`
`*c = a + *c;`

j) `a = 1;` **correto**
`c = &a;`
`for (b=0; b<10; b++) {`
`*c += 1;`
`}`

k) `c = &b;` **correto**
`b = 10;`
`printf("%d", *c);`

l) `scanf("%d", c);` **correto**

m) `a = 100;` **correto**
`c = &a;`
`while (a > 0) {`
`*c = *c - 1;`
`printf("%d\n", a);`
`}`

n) **Correto. Porém não se sabe para onde "c" está apontando.**

`*c = 10;`
`d = c;`
`*d = *d + 10;`
`printf("%d", *d);`

- 2) Analise as seguintes sequências de código. Verifique se estão corretas ou incorretas. Caso estejam incorretas, apontem os prováveis erros. Para tanto, considere a declaração das variáveis abaixo:

```
int v[5], *pi, *pa, x, i;
```

a) `pi = v;`
`*pi++ = 200;`
`pi = pi + 2;`
`*pi = 100;`
Vetor: 200 0 0 100 0

b) `v[0] = 10;`
`v[4] = 30;`
`pi = &v[4];`
`pa = &v[0];`
`*pa += *pi;`
Vetor: 40 0 0 0 30

c) `*v = 10;`
`v++;` **Erro. Não pode alterar o vetor**
`*v = 20;`

d) `scanf("%d", v);` **posição v[0]**
`if (v[0] > 10) {`
`pi = v+1;`
`*pi = 2*(*v);`
`} else {`
`*(pi+1) = v[0];`
`}`
Se ler 15 (> 10) → 15 30 0 0 0
Se ler < 10 → Erro. Não se sabe para onde "pi" aponta.

e) `scanf("%d", v+1);` **posição v[1]**
`v++;` **Erro. Retirar**
`x = *(v+1);`
`if (x < 0) {`
`pa = v;`
`for (i=0; i<5; i++) {`
`*(pa+1) = i;`
`}`
`} else {`
`pa = &v[4];`
`*pa = x + 5;`
`}`
Se ler -1 (< 0) → 0 4 0 0 0
Se ler 1 (> 0) → 0 1 0 0 6

f) `pi = v;`
`for (i=0; i<5; i++) {`
`scanf("%d", (pi+i));`
`}`
`if (*pi > *(pi+1)) {`
`x = *pi;`
`*pi = *(pi+1);`
`*(pi+1) = x;`
`}`
Se digitar 1 2 3 4 5 → v:1 2 3 4 5
Se digitar 2 1 3 4 5 → v:1 2 3 4 5

g) `v[0] = 70;`
`v[1] = 40;`
`pi = v;`
`pa = v+1;`
`if (*pi > *pa) {`
`pi = pa;`
`pa = v;`
`printf("%d %d", *pi, *pa);`
`}`
Vetor: 70 40 0 0 0
Printf: 40 70

h) `pi = &v[4];`
`v = pi-1;` **Erro. Não pode alterar o v**
`*(pi-1) = 10;` **v[3] = 10**
`printf("%d", *v);` **Imprime v[0]**

i) `pi = &x;`
`pa = v;`
`scanf("%d", &x);`
`*pa = *pi;`
`*(pa+1) = *pi-1;`
`*(pa+2) = *pi-2;`
`*(pa+3) = *pi-3;`

Se digitar 10 → Vetor: 10 9 8 7 6