

M10L: Ponteiros

Murilo Dantas

1. Para cada item abaixo, crie uma “execução” em seu caderno e, em seguida, gere um programa para conferir se seu raciocínio está correto. Sugiro que use o depurador.

a.

```
void main()
{
    int x, *pX;
    x = 10;
    pX = &x;

    printf("x = %d\n",x);
    printf("*pX = %d\n",*pX);
    printf("pX = %p\n",pX);
    printf("&x = %p\n",&x);
}
```

b.

```
void main()
{
    int x, *pX;

    pX = &x;
    x = 10;

    printf("x = %d\n",x);
    printf("*pX = %d\n",*pX);
    printf("pX = %p\n",pX);
    printf("&x = %p\n",&x);
}
```

c.

```
void main()
{
    int x, *pX;

    pX = &x;
    *pX = 15;

    printf("x = %d\n",x);
    printf("*pX = %d\n",*pX);
    printf("pX = %p\n",pX);
    printf("&x = %p\n",&x);
}
```

d.

```
void main()
{
    int x, y, *p1, *p2;

    p1 = &x;
    p2 = &y;
    x = 25;
    y = (int)sqrt(x);
    printf("x = %d\n",x);
    printf("*p1 = %d\n", *p1);
    printf("y = %d\n",y);
    printf("*p2 = %d\n", *p2);
}
```

e.

```
void main()
{
    int *p1, a = 3, b = 4, *p2;

    p1 = &a;
    p2 = &b;
    *p1 = (int)sqrt(pow(a,2)+pow(*p2,2));
    b = *p1 * b;
    p2 = p1;

    printf("*p1 = %d\n", *p1);
    printf("*p2 = %d\n", *p2);
    printf("a = %d\n",a);
    printf("b = %d\n",b);
}
```

f.

```
void main()
{
    int x, y, *p1, *p2;

    p1 = &x;
    p2 = p1;
    x = 25;
    y = (int)sqrt(x);

    printf("x = %d\n",x);
    printf("*p1 = %d\n", *p1);
    printf("y = %d\n",y);
    printf("*p2 = %d\n", *p2);
}
```

g.

```
void main()
{
    int x, y, *p1, *p2;

    p1 = &x;
    p2 = p1;
    *p2 = 25;
    y = (int)sqrt(x);

    printf("x = %d\n",x);
    printf("*p1 = %d\n",*p1);
    printf("y = %d\n",y);
    printf("*p2 = %d\n",*p2);
}
```

h.

```
void main()
{
    int x, y, *p1, *p2;

    p1 = &y;
    p2 = p1;
    *p2 = 100;
    x = (int)sqrt(*p1);

    printf("x = %d\n",x);
    printf("*p1 = %d\n",*p1);
    printf("y = %d\n",y);
    printf("*p2 = %d\n",*p2);
}
```

i.

```
void main()
{
    float x, y, *p1, *p2;
    //O que será impresso se o usuário digitar 15? Ou 30? Ou -10?
    printf("Forneca um numero: ");
    scanf("%f",&x);
    y = 10*x;
    p1 = &y;
    p2 = &x;
    y = (float)sqrt(*p1);
    printf("x = %f\n",x);
    printf("*p1 = %f\n",*p1);
    printf("y = %f\n",y);
    printf("*p2 = %f\n",*p2);
}
```

j.

```
void main()
{
    float x, y, *p1, *p2;

    p1 = &y;
    p2 = &x;
    //O que será impresso se o usuário digitar 10 e 20? Ou 3 e -8?
    printf("Forneca um numero: ");
    scanf("%f", p1);

    printf("Forneca outro numero: ");
    scanf("%f", p2);

    y = *p1+x;
    *p2 = 2*y;
    printf("x = %f\n",x);
    printf("*p1 = %f\n",*p1);
    printf("y = %f\n",y);
    printf("*p2 = %f\n",*p2);
}
```

k.

```
void main()
{
    float x, y, *p1, *p2;

    p1=&y;
    p2=&x;
    //O que será impresso se o usuário digitar 10 e 20? Ou 3 e -8?
    printf("Forneca um numero: ");
    scanf("%f", p1);

    printf("Forneca outro numero: ");
    scanf("%f", p2);

    y = *p1+*p2;
    *p2 = *p1*10;
    printf("x = %f\n",x);
    printf("*p1 = %f\n",*p1);
    printf("y = %f\n",y);
    printf("*p2 = %f\n",*p2);
}
```

I.

```
void main()
{
    char letra1 = 'x', letra2 = 'A', letra3 = '9';
    char *p1, *p2, *p3;

    p1 = &letra1;
    p2 = &letra2;
    p3 = p1;

    if(isalnum(*p1))
        printf("*p1 eh um caracter alfanumerico\n");
    if(isalpha(*p2))
        printf("*p2 eh um caracter alfabetico\n");
    if(isdigit(*p3))
        printf("*p3 eh um digito\n");

    *p3 = letra3;
    if(isdigit(*p1))
        printf("*p1 eh um digito\n");
    else if(islower(*p1))
        printf("*p1 eh um caracter minusculo\n");
    if(isdigit(*p3))
        printf("*p3 eh um digito\n");

    letra3 = ',';
    if(ispunct(*p3))
        printf("*p3 eh um sinal de pontuacao\n");
    if(p1 == &letra3)
        printf("p1 aponta para letra3\n");
    if(p3 == &letra3)
        printf("p3 aponta para letra3\n");

    letra2 = letra1;
    if(p1 == p2)
        printf("p1 e p2 apontam para o mesmo lugar\n");
    if(p1 == p3)
        printf("p1 e p3 apontam para o mesmo lugar\n");
    if(*p1 == letra3)
        printf("*p1 e letra3 tem os mesmos valores\n");
    if(p1 == p2 && p2 == p3)
        printf("p1, p2 e p3 apontam para o mesmo endereco\n");
    if(*p1 == *p2 && *p2 == *p3)
        printf("*p1, *p2 e *p3 contem os mesmos valores\n");
    if(letra1 == letra2 && letra2 == letra3)
        printf("letra1, letra2 e letra3 tem os mesmos valores\n");
}
```

2. Para cada item abaixo, crie uma “execução” em seu caderno e, em seguida, gere um programa para conferir se seu raciocínio está correto. Sugiro que use o depurador.

a.

```
void main()
{
    int i = 3, v[] = {10,15,20,30,35,40}, *pV;
    pV = v;
    printf("*pV: %d\n", *pV);
    printf("*pV+1: %d\n", *pV+1);
    printf("(*(pV+1): %d\n", *(pV+1));
    printf("(*(pV+3): %d\n", *(pV+3));
    printf("(*(pV+i): %d\n", *(pV+i));
    printf("*pV + --i: %d\n", *pV + --i);
    printf("(*(pV + --i): %d\n", *(pV + --i));
    printf("(*(pV+i+2): %d\n", *(pV+i+2));
    printf("(*(pV+i-1): %d\n", *(pV+i-1));
    if(pV == v)
        printf("pV aponta para o inicio de v.\n");
    else
        printf("pV nao aponta para o inicio de v.\n");
    getch(); //Pesquise o que faz essa função!

    pV++;
    printf("*pV: %d\n", *pV);
    printf("*pV+1: %d\n", *pV+1);
    printf("(*(pV+1): %d\n", *(pV+1));
    printf("(*(pV+3): %d\n", *(pV+3));
    printf("(*(pV+i): %d\n", *(pV+i));
    printf("*pV + --i: %d\n", *pV + --i);
    printf("(*(pV + --i): %d\n", *(pV + --i));
    printf("(*(pV+i+2): %d\n", *(pV+i+2));
    printf("(*(pV+i-1): %d\n", *(pV+i-1));
    if(pV == v)
        printf("pV aponta para o inicio de v.\n");
    else
        printf("pV nao aponta para o inicio de v.\n");
    getch();

    pV += 4;
    printf("*pV: %d\n", *pV);
    printf("*pV+1: %d\n", *pV+1);
    printf("(*(pV+1): %d\n", *(pV+1));
    printf("(*(pV+3): %d\n", *(pV+3));
    printf("(*(pV+i): %d\n", *(pV+i));
    printf("*pV + --i: %d\n", *pV + --i);
    printf("(*(pV + --i): %d\n", *(pV + --i));
    printf("(*(pV+i+2): %d\n", *(pV+i+2));
    printf("(*(pV+i-1): %d\n", *(pV+i-1));
```

```

if(pV == v)
    printf("pV aponta para o inicio de v.\n");
else
    printf("pV nao aponta para o inicio de v.\n");
getch();

pV--;
printf("**pV: %d\n", *pV);
printf("**pV+1: %d\n", *pV+1);
printf("** (pV+1): %d\n", *(pV+1));
printf("** (pV+3): %d\n", *(pV+3));
printf("** (pV+i): %d\n", *(pV+i));
printf("**pV + --i: %d\n", *pV + --i);
printf("** (pV + --i): %d\n", *(pV + --i));
printf("** (pV+i+2): %d\n", *(pV+i+2));
printf("** (pV+i-1): %d\n", *(pV+i-1));
if(pV == v)
    printf("pV aponta para o inicio de v.\n");
else
    printf("pV nao aponta para o inicio de v.\n");
}

```

b.

```

void main()
{
    //Execute para os valores:
    //10, 20, 30, 40, 50 (vetor) e 3 (i)
    //-25.8, 12.3, 15, -10.57, -8.9 (vetor) e -1 (i)
    float *pVf, vf[5], expr;
    int i;
    for(i=0;i<5;i++)
    {
        printf("\nv[%d] = ",i);
        scanf("%f",&vf[i]);
    }
    printf("i = ");
    scanf("%d",&i);

    pVf = vf;
    expr = *pVf+pVf[i+1]+*(vf+i-2);
    printf("A expressao *pVf+pVf[i+1]+*(vf+i-2) vale %f\n", expr);
}

```

c.

```

void main()
{
    char *pc,vc1[] = "Informatica";
    char vc2[11];

```

```

int i;

pc = vc1+5;

for(i=0;i<7;i++)
    vc2[i+4] = pc[i];
vc2[0] = toupper(*(vc1+i-2));
vc2[1] = pc[5];
*(vc2+2) = *(pc+2);
pc = vc2;
pc[3] = 'e';

printf("vc2: %s\n",vc2);
}

```

d.

```

void main()
{
    char *pi, *pm, *pf;
    char s1[] = "Algoritmos Estruturados";
    char s2[19];
    int i;

    pi = s2;
    pm = s2+10;
    pf = pm+8;
    for(i=0; i<10; i++)
        pm[i-10] = *(s1+i+11);
    i = (int)i*1.5;
    *(s2+i) = 'd';
    pm[-1] = *(s1+i+8);
    *(pf-1) = *(s1+9);
    pm[3] = pm[0] = pm[5];
    *(pf-6) = pf[-9];
    pi[14] = tolower(*s1);
    s2[16] = s1[21];
    *(pm+1) = tolower(s1[11]);
    pf[0] = *(s1+23);

    printf("String: %s\n", pf-18);
}

```


e.

```
void main()
{
    int *p1,*p2, i;
    int v[] = {15,28,34,-15,12,-9};
    int exp1, exp2;

    p1 = v;
    p2 = p1+2;
    for(i=0;i<4;i++)
    {
        exp1 = (*p1*10)+p2[3-i]*5;
        exp2 = *(++p1)+*(p2+i-1);
        printf("exp1: %d\n", exp1);
        printf("exp2: %d\n", exp2);

        if(p1 > p2)
            printf("p1 aponta para um endereço maior que p2\n");
        else if(p1 == p2)
            printf("p1 e p2 apontam para o mesmo endereço\n");
        else
            printf("p2 aponta para um endereço maior que p1\n");
        getche(); //Pesquise o que faz essa função!
    }
}
```

f.

```
void main()
{
    int i, *pi, **ppi;
    pi = &i;
    ppi = &pi;
    i = 10;
    printf("i = %d\n*pi = %d\n**ppi = %d\n",i,*pi,**ppi);
}
```

g.

```
void main(){
    double **pp, *pd, num=34;
    pd = &num;
    pp = &pd;
    (**pp)++;
    num = **pp + *pd;

    printf("num = %lf\n",num);
    printf("**pd = %lf\n",*pd);
    printf("***pp = %lf\n",**pp);
}
```

h.

```
void main()
{
    int **pp1, **pp2, num1, num2;
    int *pi, aux;

    num1 = 10;
    num2 = 20;
    pi = &aux;
    pp1 = &pi;
    pp2 = &pi;

    *pi = num1 + num2;
    num1 = **pp1+5;
    num2 = (**pp2)--;

    printf("num1 = %d\n", num1);
    printf("num2 = %d\n", num2);
    printf("*pi = %d\n", *pi);
    printf("**pp1 = %d\n", **pp1);
    printf("**pp2 = %d\n", **pp2);
}
```

i.

```
void main()
{
    char **ppc, *pc, cad[] = "algoritmos";
    pc = cad;
    ppc = &pc;
    *(*ppc+9) = '1';

    printf("pc = %s\n", pc);
    printf("cad = %s\n", cad);
    printf("**ppc = %c\n", **ppc);
    printf("*ppc = %s\n", *ppc);
}
```

- 3. Faça uma versão nova de todos os exercícios da lista sobre vetores, alterando a declaração dos vetores do modo estático usando constantes, para o modo dinâmico, usando ponteiros.**
- 4. Faça uma versão nova de todos os exercícios da lista sobre matrizes, alterando a declaração das matrizes do modo estático usando constantes, para o modo dinâmico, usando ponteiros.**