

Exemplo

```
int x = 10;

int * y = &x;

printf("\n%i", x);

printf("\n%p", &x);

printf("\n%p", y);

printf("\n%p", &y);

printf("\n%i", *y);
```

• Faça o quadro de memória e mostre a saída na tela:

```
int x = 10;

int * y = &x;

printf("\n%i", x);

printf("\n%p", &x);

printf("\n%p", y);

printf("\n%p", &y);

printf("\n%i", *y);
```

| Memória | | |
|---------|----|-----|
| x | 10 | 75h |
| | | ... |
| y | | CDh |

| Tela |
|------|
| |

• Faça o quadro de memória e mostre a saída na tela:

```
int x = 10;

int * y = &x;

printf("\n%i", x);

printf("\n%p", &x);

printf("\n%p", y);

printf("\n%p", &y);

printf("\n%i", *y);
```

| Memória | | |
|---------|-----|-----|
| x | 10 | 75h |
| | | ... |
| y | 75h | CDh |

| Tela |
|------|
| |

• Faça o quadro de memória e mostre a saída na tela:

```
int x = 10;

int * y = &x;

printf("\n%i", x);

printf("\n%p", &x);

printf("\n%p", y);

printf("\n%p", &y);

printf("\n%i", *y);
```

| Memória | | |
|---------|-----|-----|
| x | 10 | 75h |
| | | ... |
| y | 75h | CDh |

| Tela |
|------|
| 10 |

• Faça o quadro de memória e mostre a saída na tela:

```
int x = 10;

int * y = &x;

printf("\n%i", x);

printf("\n%p", &x);

printf("\n%p", y);

printf("\n%p", &y);

printf("\n%i", *y);
```

| Memória | | |
|---------|-----|-----|
| x | 10 | 75h |
| | | ... |
| y | 75h | CDh |

| Tela |
|-----------|
| 10 75h |

• Faça o quadro de memória e mostre a saída na tela:

```
int x = 10;

int * y = &x;

printf("\n%i", x);

printf("\n%p", &x);

printf("\n%p", y);

printf("\n%p", &y);

printf("\n%i", *y);
```

| Memória | | |
|---------|-----|-----|
| x | 10 | 75h |
| | | ... |
| y | 75h | CDh |

| Tela |
|------------------|
| 10 75h 75h |

• Faça o quadro de memória e mostre a saída na tela:

```
int x = 10;

int * y = &x;

printf("\n%i", x);

printf("\n%p", &x);

printf("\n%p", y);

printf("\n%p", &y);

printf("\n%i", *y);
```

| Memória | | |
|---------|-----|-----|
| x | 10 | 75h |
| | | ... |
| y | 75h | CDh |

| Tela |
|-------------------------|
| 10 75h 75h CDh |

• Faça o quadro de memória e mostre a saída na tela:

```
int x = 10;

int * y = &x;

printf("\n%i", x);

printf("\n%p", &x);

printf("\n%p", y);

printf("\n%p", &y);

printf("\n%i", *y);
```

| Memória | | |
|---------|-----|-----|
| x | 10 | 75h |
| | | ... |
| y | 75h | CDh |

| Tela |
|-------------------------------|
| 10 75h 75h CDh 10 |

Exemplo

```

int x1, x2, x3; int *p;
x1 = 11; x2 = 22; x3 = 33;
p = &x1;
x2 = *p;
*p = x3;
p = &x3;
*p = 0;

printf("cont:%d %d %d %d", x1, x2, x3, *p);
printf("addr:%p %p %p %p", &x1, &x2, &x3, p);

```

```

int x1, x2, x3; int *p;
x1 = 11; x2 = 22; x3 = 33;
p = &x1;
x2 = *p;
*p = x3;
p = &x3;
*p = 0;

printf("cont:%d %d %d %d", x1, x2, x3, *p);
printf("addr:%p %p %p %p", &x1, &x2, &x3, p);

```

| Memória | | |
|---------|---|------|
| x1 | ? | 89Bh |
| x2 | ? | 89Ch |
| x3 | ? | 89Dh |
| p | ? | 89Eh |

Algoritmos e Estruturas de Dados II (R2)

```

int x1, x2, x3; int *p;
x1 = 11; x2 = 22; x3 = 33;
p = &x1;
x2 = *p;
*p = x3;
p = &x3;
*p = 0;

printf("cont:%d %d %d %d", x1, x2, x3, *p);
printf("addr:%p %p %p %p", &x1, &x2, &x3, p);

```

| Memória | | |
|---------|----|------|
| x1 | 11 | 89Bh |
| x2 | 22 | 89Ch |
| x3 | 33 | 89Dh |
| p | ? | 89Eh |

Algoritmos e Estruturas de Dados II (R2)

```

int x1, x2, x3; int *p;
x1 = 11; x2 = 22; x3 = 33;
p = &x1;
x2 = *p;
*p = x3;
p = &x3;
*p = 0;

printf("cont:%d %d %d %d", x1, x2, x3, *p);
printf("addr:%p %p %p %p", &x1, &x2, &x3, p);

```

| Memória | | |
|---------|------|------|
| x1 | 11 | 89Bh |
| x2 | 22 | 89Ch |
| x3 | 33 | 89Dh |
| p | 89Bh | 89Eh |

Algoritmos e Estruturas de Dados II (R2)

```

int x1, x2, x3; int *p;
x1 = 11; x2 = 22; x3 = 33;
p = &x1;
x2 = *p;
*p = x3;
p = &x3;
*p = 0;

printf("cont:%d %d %d %d", x1, x2, x3, *p);
printf("addr:%p %p %p %p", &x1, &x2, &x3, p);

```

| Memória | | |
|---------|------|------|
| x1 | 11 | 89Bh |
| x2 | 11 | 89Ch |
| x3 | 33 | 89Dh |
| p | 89Bh | 89Eh |

Algoritmos e Estruturas de Dados II (R2)

```

int x1, x2, x3; int *p;
x1 = 11; x2 = 22; x3 = 33;
p = &x1;
x2 = *p;
*p = x3;
p = &x3;
*p = 0;

printf("cont:%d %d %d %d", x1, x2, x3, *p);
printf("addr:%p %p %p %p", &x1, &x2, &x3, p);

```

| Memória | | |
|---------|------|------|
| x1 | 33 | 89Bh |
| x2 | 11 | 89Ch |
| x3 | 33 | 89Dh |
| p | 89Bh | 89Eh |

Algoritmos e Estruturas de Dados II (R2)

```

int x1, x2, x3;  int *p;
x1 = 11;  x2 = 22;  x3 = 33;

p = &x1;
x2 = *p;
*p = x3;
p = &x3;
*p = 0;

printf("cont:%d %d %d %d", x1, x2, x3, *p);
printf("addr:%p %p %p %p", &x1, &x2, &x3, p);

```

| Memória | | |
|---------|------|------|
| x1 | 33 | 89Bh |
| x2 | 11 | 89Ch |
| x3 | 33 | 89Dh |
| p | 89Dh | 89Eh |

```

int x1, x2, x3;  int *p;
x1 = 11;  x2 = 22;  x3 = 33;
p = &x1;
x2 = *p;
*p = x3;
p = &x3;


*p = 0;



printf("cont: %d %d %d %d", x1, x2, x3, *p);
printf("addr: %p %p %p %p", &x1, &x2, &x3, p);

```

| Memória | | |
|---------|------|------|
| x1 | 33 | 89Bh |
| x2 | 11 | 89Ch |
| x3 | 0 | 89Dh |
| p | 89Ch | 89Eh |

```
int x1, x2, x3;  int *p;
x1 = 11;  x2 = 22;  x3 = 33;

p = &x1;
x2 = *p;
*p = x3;
p = &x3;
*p = 0;
```

```
printf("cont:%d %d %d %d", x1, x2, x3, *p);
printf("addr:%p %p %p %p", &x1, &x2, &x3, p);
```

| Memória | | |
|---------|------|------|
| x1 | 33 | 896h |
| x2 | 11 | 89Ch |
| x3 | 0 | 89Dh |
| p | 89Dh | 89Eh |

Cont.33 11 0 0

```

int x1, x2, x3;  int *p;

x1 = 11;  x2 = 22;  x3 = 33;

p = &x1;
x2 = *p;

*p = x3;
p = &x3;
*p = 0;

printf("cont %d %d %d %d", x1, x2, x3, *p);
printf("addr %p %p %p %p", &x1, &x2, &x3, p);

```

| Memória | |
|---------|------|
| x1 | 33 |
| x2 | 11 |
| x3 | 0 |
| p | 89Ch |

Cont: 33 11 0 0
 Cam: 89Bh 89Ch 89Ch 89Dh

Exemplo

```
int *x1;      int x2;      int *x3;

x1 = (int *) malloc (sizeof(int));
printf("x1[%p] %i %p x2[%p] x3[%p] %p", x1, *x1, &x1, x2, &x2, x3, &x3);

*x1 = 20;
printf("x1[%p] %i %p x2[%p] x3[%p] %p", x1, *x1, &x1, x2, &x2, x3, &x3);

x2 = *x1;
printf("x1[%p] %i %p x2[%p] x3[%p] %p", x1, *x1, &x1, x2, &x2, x3, &x3);

*x3 = x2 * *x1;
printf("x1[%p] %i %p x2[%p] x3[%p] %p", x1, *x1, &x1, x2, &x2, x3, &x3);

x3 = &x2;
printf("x1[%p] %i %p x2[%p] x3[%p] %p", x1, *x1, &x1, x2, &x2, x3, &x3);

x2 = 15;
printf("x1[%p] %i %p x2[%p] x3[%p] %p", x1, *x1, &x1, x2, &x2, x3, &x3);
```

[illegible][illegible]

| reg r1, r2, r3 | reg r2, r3, r4 | reg r3, r4, r5 |
|---|----------------|----------------|
| <pre> r1 = r2 + r3; printf("r1=%d\n", r1); r2 = r1 + r2; printf("r2=%d\n", r2); r3 = r2 + r3; printf("r3=%d\n", r3); r4 = r3 + r4; printf("r4=%d\n", r4); r5 = r4 + r5; printf("r5=%d\n", r5); </pre> | | |
| r1 | A3h | A0h |
| r2 | ? | A1h |
| r3 | ? | A2h |

| Test |
|-------------------------|
| 1) A3h(A0hA0h) A2h(A1h) |
| 2) A2h(A1hA2h) |
| |
| |
| |

| | | |
|---------------------------|---------|-----|
| | Memória | |
| x1 | A3h | A0h |
| x2 | ? | A1h |
| x3 | ? | A2h |
| Tela | | |
| [FACIL] [MÉDIO] [DIFÍCIL] | | |
| [QUIZ] [TEMA] | | |
| | | |
| | | |
| | | |

| | Memoria | |
|----|---------|-----|
| x1 | A3h | A0h |
| x2 | ? | A1h |
| x3 | ? | A2h |

| Tota | |
|----------------------|--|
| #1A0B:A2hA0h: A2hA1h | |
| A0A2h:A5A2h | |
| #1A0B:A2hA0h: A2hA1h | |
| A0A2h:A5A2h | |
| | |
| | |

| | Memória | |
|----|---------|-----|
| x1 | A2h | A0h |
| x2 | 20 | A1h |
| x3 | ? | A2h |

| Tela | |
|---------------------|--|
| x1A0hA2h0A0hA2h0A0h | |
| x3A2h7(A2h) | |
| x1A0hA2h0A0hA2h0A0h | |
| x3A2h7(A2h) | |

| | | |
|----|---------|-----|
| | Memória | |
| x1 | A3h | A0h |
| x2 | 20 | A1h |
| x3 | ? | A2h |

| |
|--|
| Tela |
| +1A0h (+200Ah) (+270Ah) +5A2h (+7A2h) |
| +1A0h (+200Ah) (+270Ah) +5A2h (+7A2h) |
| +1A0h (+200Ah) (+270Ah) +5A2h (+7A2h) |

| | Memoria | |
|----|---------|-----|
| x1 | A3h | A0h |
| x2 | 20 | A1h |
| x3 | 400 | A2h |

| Tela | |
|------------|---------|
| x1A0x20A0h | x275A0h |
| x3A0h | 75A0h |
| x1A0x20A0h | x275A0h |
| x3A0h | 75A0h |
| x1A0x20A0h | x275A0h |
| x3A0h | 75A0h |

| | Memória | |
|----|---------|-----|
| x1 | A3h | A0h |
| x2 | 20 | A1h |
| x3 | 400 | A2h |

| Tela |
|---------------------------------------|
| x10A9A39 6A00 4378A1h x3A2h=7A2h |
| x7A9B67E 9A00 4378A1h x3A2h=7A2h |
| x7A9B67E 9A00 4378A1h x3A2h=400A2h |

```

x1 = 10;      x2 = 20;      x3 = 30;

x1 = (x1 * 2) + (x2 * 3) * x3;
printf("x1=%d, x2=%d, x3=%d\n", x1, x2, x3);

x1 = 20;
printf("x1=%d, x2=%d, x3=%d\n", x1, x2, x3);

x2 = 30;
printf("x1=%d, x2=%d, x3=%d\n", x1, x2, x3);

x3 = x2 * 2;
printf("x1=%d, x2=%d, x3=%d\n", x1, x2, x3);

x1 = x3;
printf("x1=%d, x2=%d, x3=%d\n", x1, x2, x3);

```

| | Memoria | |
|----|---------|-----|
| x1 | A3h | A0h |
| x2 | 20 | A1h |
| x3 | A1h | A2h |

| Tela | |
|------|----|
| x1 | 10 |
| x2 | 20 |
| x3 | 30 |

The diagram illustrates the execution of a loop in assembly. The left side shows the assembly code with comments. The right side shows the state of registers x1, x2, and x3 at each iteration of the loop.

Assembly Code:

```

x1 = 0; // register initialized
printf("while loop\n");
while(1) {
    x1 = x1 + 1;
    printf("x1=%d\n", x1);
    x2 = x2 + 1;
    printf("x2=%d\n", x2);
    x3 = x3 + 1;
    printf("x3=%d\n", x3);
    if(x1 == 10) {
        printf("while loop finished\n");
        break;
    }
}

```

Register State:

| | Memória | |
|----|---------|-----|
| x1 | A3h | A3h |
| x2 | 15 | A1h |
| x3 | A1h | A2h |

Trace:

```

x1:00000000 x2:00000000 x3:00000000
x1:00000001 x2:00000001 x3:00000001
x1:00000002 x2:00000002 x3:00000002
x1:00000003 x2:00000003 x3:00000003
x1:00000004 x2:00000004 x3:00000004
x1:00000005 x2:00000005 x3:00000005
x1:00000006 x2:00000006 x3:00000006
x1:00000007 x2:00000007 x3:00000007
x1:00000008 x2:00000008 x3:00000008
x1:00000009 x2:00000009 x3:00000009
x1:0000000A x2:0000000A x3:0000000A

```

```

double M [12]
double *p = M[5]
for (int i = 0; i < pow(MAXTAM, 2); i += p++) {
    *p = 0.0;
}

```

Exercício

• Mostre a saída na tela

```
double a;
double *p, *q;
a = 3.14;
printf("%f\n", a);
p = &a;
*p = 2.718;
printf("%f\n", a);
a = 5;
printf("%f\n", *p);

p = NULL;
p = (double*) malloc(sizeof(double));
*p = 20;
q = p;
printf("%f\n", *p);
printf("%f\n", a);
free(p);
printf("%f\n", *q);
```

3.14
2.718
5
20
5
NULL

Exercício

• Mostre o quadro de memória

```
int a[10];
a = 3;
a[5] = 100;
printf("%d\n", a[5]);

a = (int*) malloc(10*sizeof(int));
a[7] = 100;
printf("%d\n", a[7]);

//O comando a = b gera um erro de compilação
```

Exercício

```
int *x1; int *x2; int *x3;
x1 = (int*) malloc(sizeof(int));
*x1 = 20;
x2 = *x1;
*x3 = x2 + *x1;
x3 = &x2;
x2 = 15;
x2 = 13 & 5;
x2 = 13 | 5;
x2 = 13 ^ 5;
x2 = 13 >> 1;
x2 = 13 << 1;
```

```
int *x1; int *x2; int *x3;
x1 = (int*) malloc(sizeof(int));
*x1 = 30;
x2 = *x1;
*x3 = x2 + *x1;
x3 = &x2;
x2 = 15;
x2 = 13 & 5;
x2 = 13 | 5;
x2 = 13 ^ 5;
x2 = 13 >> 1;
x2 = 13 << 1;
```

Exercício

