

PONTEIROS

```
int main(void){
    int num = 55, valor;
    int *p;
    p = &num;
    valor = *p;
    cout << valor;
    cout << "para onde o ponteiro aponta: " << p;
    cout << "Valor da variavel apontada: " << *p << " ";
    return(0);
}
```

```
55
para onde o ponteiro aponta: 0x28ff04
Valor da variavel apontada: 55
```

```
int main (void) {
    int cont=10;
    int *pt;
    pt = &cont;

    cout << "&pt= " << &pt << &pt+1 << &pt-1 << &pt-2 << endl;
    cout << "pt= " << pt << endl;
    cout << "*pt= " << *pt << endl;
    cout << "&pt= " << &(*pt) <<endl;
    return 0 ;
}
```

```
&pt= 0x28ff08 0x28ff0c 0x28ff04 0x28ff00
pt= 0x28ff0c
*pt= 10
&pt= 0x28ff0c
```

using namespace std;

```
int main(void){
    double *p; // vale tambem para int ou float
    cout << "Endereco de p (&p): " << &p << endl;
    cout << "Endereco de p (&p)+1: " << &p+1 << endl;
    cout << "Endereco de p (&p)+2: " << &p+2 << endl;
    cout << "Endereco de p (&p)+3: " << &p+3 << endl;
    cout << "Endereco de p (&p)-1: " << &p-1 << endl;
    cout << "Endereco de p (&p)-2: " << &p-2 << endl;
    return (0);
}
```

```
Endereço de p (&p): 0x28ff0c
Endereço de p (&p)+1: 0x28ff10
Endereço de p (&p)+2: 0x28ff14
Endereço de p (&p)+3: 0x28ff18
Endereço de p (&p)-1: 0x28ff08
Endereço de p (&p)-2: 0x28ff04
```

```
int main(void){
    int *p1,*p2,x=17;
    p1 = &x;
    p2 = p1++;
    cout << "& de p1: " << &p1 << endl;
    cout << "Conteudo de p1: " << p1 << endl;
    cout << "Valor apontado por p1: " << *p1 << endl;
    cout << endl;

    cout << "& de p2: " << &p2 << endl;
    cout << "Conteudo de p2: " << p2 << endl;
    cout << "Valor apontado por p2: " << *p2 << endl;

    return (0);
}
```

```
& de p1: 0x28ff0c
Conteudo de p1: 0x28ff08
Valor apontado por p1: 2686724

& de p2: 0x28ff08
Conteudo de p2: 0x28ff04
Valor apontado por p2: 17
```

PONTEIROS

```
#include <iostream>
```

using namespace std;

```
int main(void){
    int p[5]={10,9,8,7,6};

    cout << *p << endl;
    cout << *(p+1) << endl;
    cout << *(p+2) << endl;
    cout << *(p+3) << endl;
    cout << *(p+4) << endl;
    cout << *(p+5) << endl;    ///Lixo????
}
```

```
10
9
8
7
6
4273104
```

```

    return (0);
}

int main(void){
    int num = 55, valor;
    int *p;
    p = &num;
    valor = *p;
    cout << valor;
    cout << "para onde o ponteiro aponta: " << p;
    cout << "Valor da variavel apontada: " << *p << " ";
    return(0);
}

```

```

55
para onde o ponteiro aponta: 0x28ff04
Valor da variavel apontada: 55

```

```

int main (void) {
    int cont=10;
    int *pt;
    pt = &cont;

    cout << "&pt= " << &pt << " &pt+1 << &pt-1 << &pt-2 << endl;
    cout << "pt= " << pt << endl;
    cout << "*pt= " << *pt << endl;
    cout << "&pt= " << &(*pt) <<endl;
    return 0 ;
}

```

```

&pt= 0x28ff08 0x28ff0c 0x28ff04 0x28ff00
pt= 0x28ff0c
*pt= 10
&pt= 0x28ff0c

```

```
using namespace std;
```

```

int main(void){
    double *p; // vale tambem para int ou float
    cout << "Endereco de p (&p): " << &p << endl;
    cout << "Endereco de p (&p)+1: " << &p+1 << endl;
    cout << "Endereco de p (&p)+2: " << &p+2 << endl;
    cout << "Endereco de p (&p)+3: " << &p+3 << endl;
    cout << "Endereco de p (&p)-1: " << &p-1 << endl;
    cout << "Endereco de p (&p)-2: " << &p-2 << endl;
    return (0);
}

```

```

Endereço de p (&p): 0x28ff0c
Endereço de p (&p)+1: 0x28ff10
Endereço de p (&p)+2: 0x28ff14
Endereço de p (&p)+3: 0x28ff18
Endereço de p (&p)-1: 0x28ff08
Endereço de p (&p)-2: 0x28ff04

```

```

int main(void){
    int *p1,*p2,x=17;
    p1 = &x;
    p2 = p1++;
    cout << "& de p1: " << &p1 << endl;
    cout << "Conteudo de p1: " << p1 << endl;
    cout << "Valor apontado por p1: " << *p1 << endl;
    cout << endl;

    cout << "& de p2: " << &p2 << endl;
    cout << "Conteudo de p2: " << p2 << endl;
    cout << "Valor apontado por p2: " << *p2 << endl;

    return (0);
}

```

```

& de p1: 0x28ff0c
Conteudo de p1: 0x28ff08
Valor apontado por p1: 2686724

& de p2: 0x28ff08
Conteudo de p2: 0x28ff04
Valor apontado por p2: 17

```