type

int x;

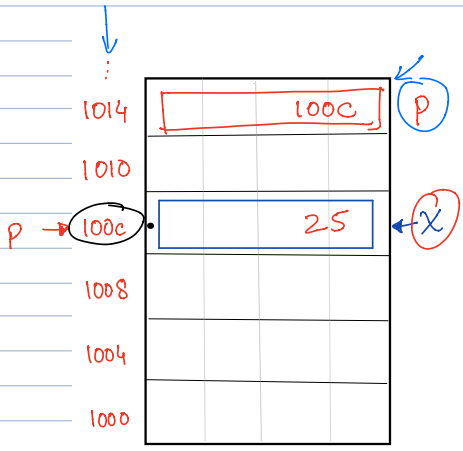
x = 25;

"address-of" operator

cout << &x ; //100C

address (number)

x ——→ x

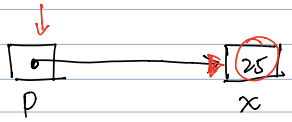


int pointer * p ; → "address"

"the thing it points to is an integer"

p = &x ;

int *p ;




```
int main() {
    int x;
    x = 25;
    int *p;
    p = &x;
    cout << p ;
    cout << *p ;
```

100C 100C

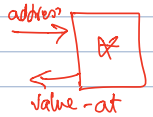
Cout << x ;

// 100C

"follow the pointer"

//25

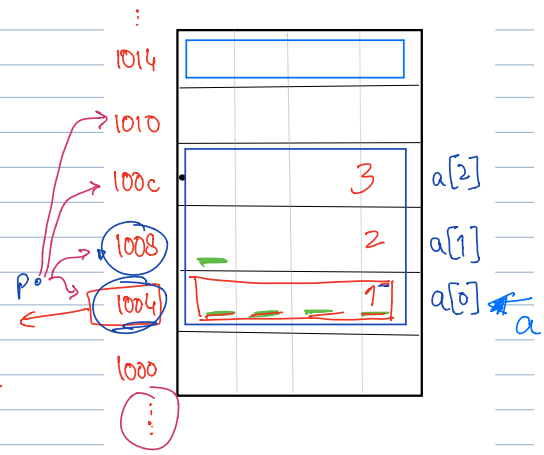
*p → 25



that address

"value-at" operator

```
int a[] = {1, 2, 3};
```




```
int *p;
```

pointer to an integer.

address of an int

```
p = &a[0];
```

address of an integer

→ a

```
cout << *p; // 1
```

```
p = p + 1; X // 1004
                1005
p = p + 4; X // 1008
```

*p → 2

int → allocated 4 bytes.

I'm pointing to an int → int = 4 bytes

```
p = p + ⌈⌉;   // 1004 → 1008
```

```
p = a;   // p = &a[0]
```

```
cout << a;
cout << a[0];
```

```
cout << x;
p = a;
```

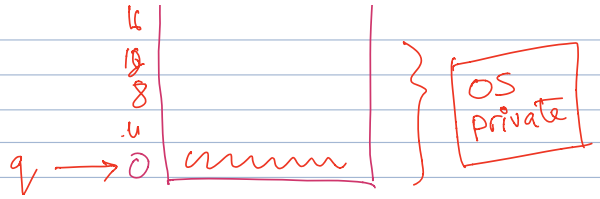
```
for (int i = 0; i < 3; i++) {
```

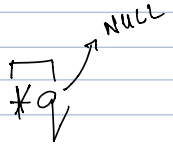
run the loop 3 times.

```
    cout << *p;
    p = p + 1;
}
```

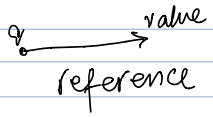
// p++;

*q





;



NULL pointer de-referencing // error / crash