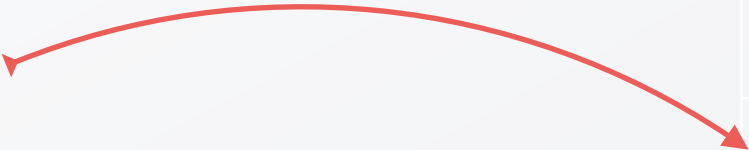


# 指针

[huiw@suda.edu.cn](mailto:huiw@suda.edu.cn)

存储器

地址



0	1010 0100
...	
800	1010 0100
801	1010 0110
802	1011 1100
803	1000 0100
804	1010 0111
...	
n-1	

## 存储器

link 的地址

```
int link = 100;
```

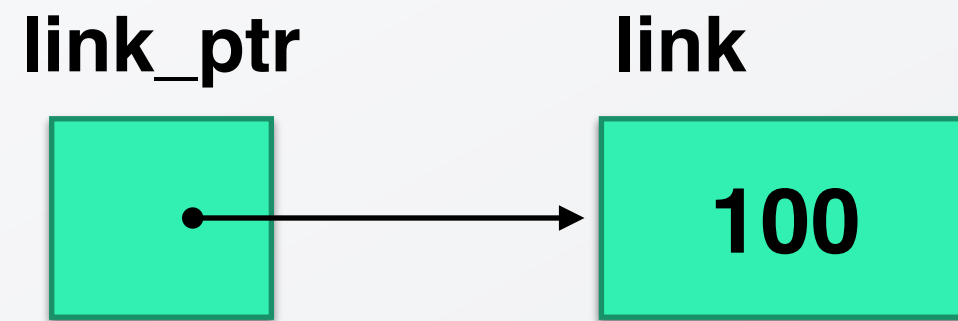
link 的内容

0	1010 0100
...	
800	0110 0100
801	0000 0000
802	0000 0000
803	0000 0000
804	1010 0111
...	
n-1	

注意：示例采用小端存储。

整数变量 **ink** 的地址，是可以存储起来的。

存储数据对象地址的变量，就是一个**指针**变量。



示例中的指针变量的名字 `link_ptr`

```
#include <stdio.h>
```

```
int link = 100;  
int *link_ptr;
```

```
int main (int argc, char* argv[]) {  
    link_ptr = &link;
```

```
    printf("value of link \t= %d\n", link);  
    printf("address of link \t= %p\n", &link);  
    printf("value of link_ptr \t= %p\n", &link_ptr);  
    printf("address of link_ptr \t= %p\n", link_ptr);
```

```
    printf("value that link_ptr is pointing = %d\n", *link_ptr);
```

```
    return 0;  
}
```

value of link = 100

address of link = 0x108857018

value of link\_ptr = 0x108857020

address of link\_ptr = 0x108857018

value that link\_ptr is pointing = 100



```
link_ptr = &link;
```

指针变量link\_ptr存储了link变量的左值。

**&**

取地址运算符

**\***

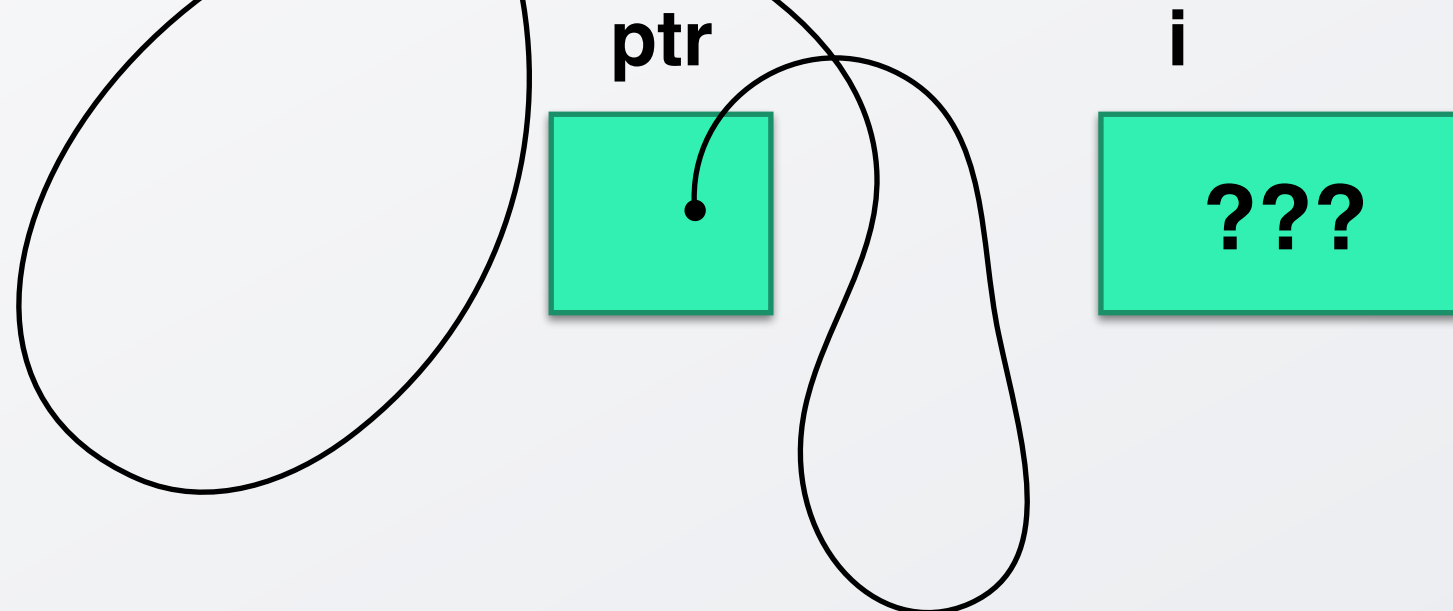
间接访问运算符

```
int *p;    /* points only to integers */
double *q; /* points only to doubles */
char *r;   /* points only to characters */
```

Might be somewhere in Cybertron.

```
int *ptr;
```

```
int i;
```



指针变量必须赋值后，才能使用。

```
#include <stdio.h>
```

```
int main(int argc, char *argv[]) {
```

```
    int i;
```

```
    int *p;
```

```
    p = &i;
```

```
    i = 1;
```

```
    printf("%d\n", i); /* prints 1 */
```

```
    printf("%d\n", *p); /* prints 1 */
```

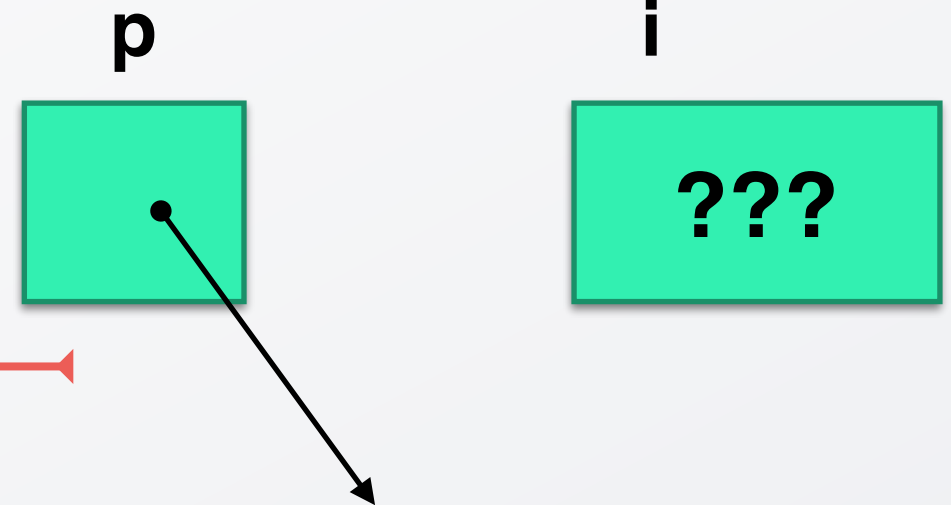
```
    *p = 2;
```

```
    printf("%d\n", i); /* prints 2 */
```

```
    printf("%d\n", *p); /* prints 2 */
```

```
    return 0;
```

```
}
```



```
#include <stdio.h>
```

```
int main(int argc, char *argv[]) {
```

```
    int i;
```

```
    int *p;
```

```
    p = &i;
```

```
    i = 1;
```

```
    printf("%d\n", i); /* prints 1 */
```

```
    printf("%d\n", *p); /* prints 1 */
```

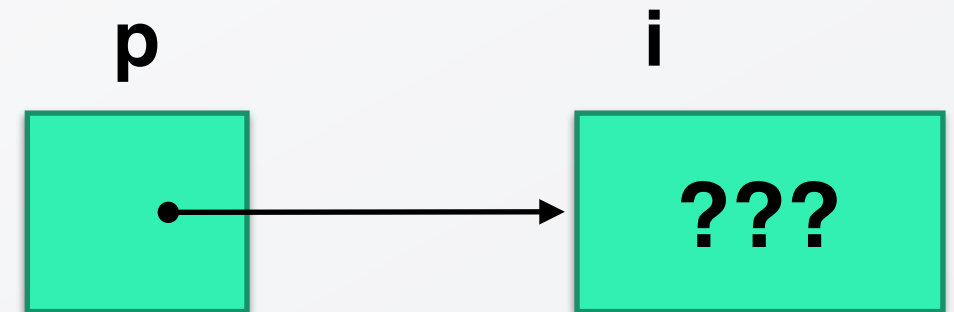
```
    *p = 2;
```

```
    printf("%d\n", i); /* prints 2 */
```

```
    printf("%d\n", *p); /* prints 2 */
```

```
    return 0;
```

```
}
```



```
#include <stdio.h>
```

```
int main(int argc, char *argv[]) {
```

```
    int i;
```

```
    int *p;
```

```
    p = &i;
```

```
    i = 1;
```

```
    printf("%d\n", i); /* prints 1 */
```

```
    printf("%d\n", *p); /* prints 1 */
```

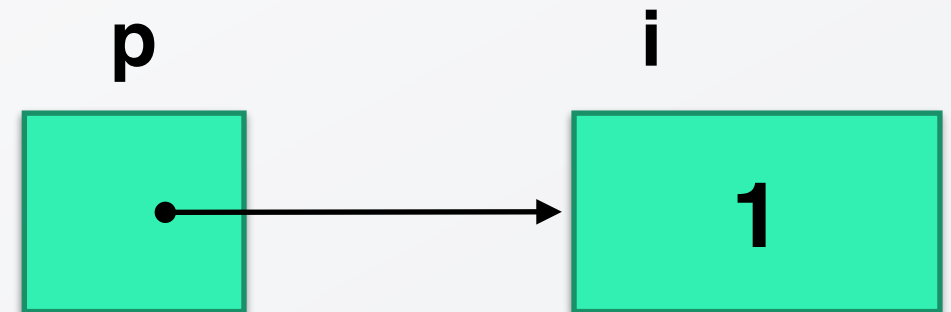
```
    *p = 2;
```

```
    printf("%d\n", i); /* prints 2 */
```

```
    printf("%d\n", *p); /* prints 2 */
```

```
    return 0;
```

```
}
```





```
#include <stdio.h>
```

```
int main(int argc, char *argv[]) {
```

```
    int i;
```

```
    int *p;
```

```
    p = &i;
```

```
    i = 1;
```

```
    printf("%d\n", i); /* prints 1 */
```

```
    printf("%d\n", *p); /* prints 1 */
```

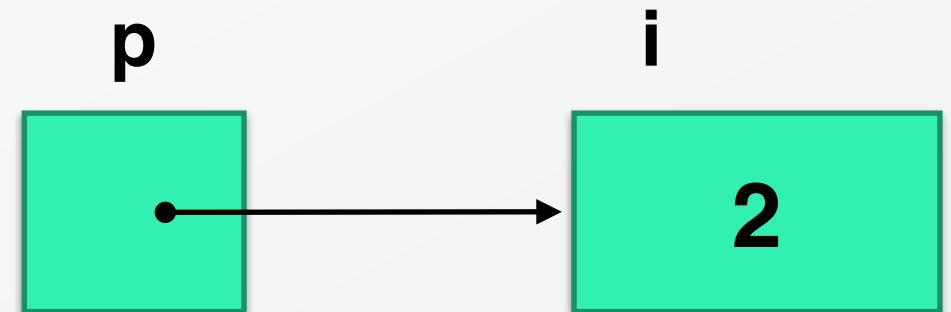
```
    *p = 2;
```

```
    printf("%d\n", i); /* prints 2 */
```

```
    printf("%d\n", *p); /* prints 2 */
```

```
    return 0;
```

```
}
```





```
void foo() {
```

```
    int i, *p, *q;
```

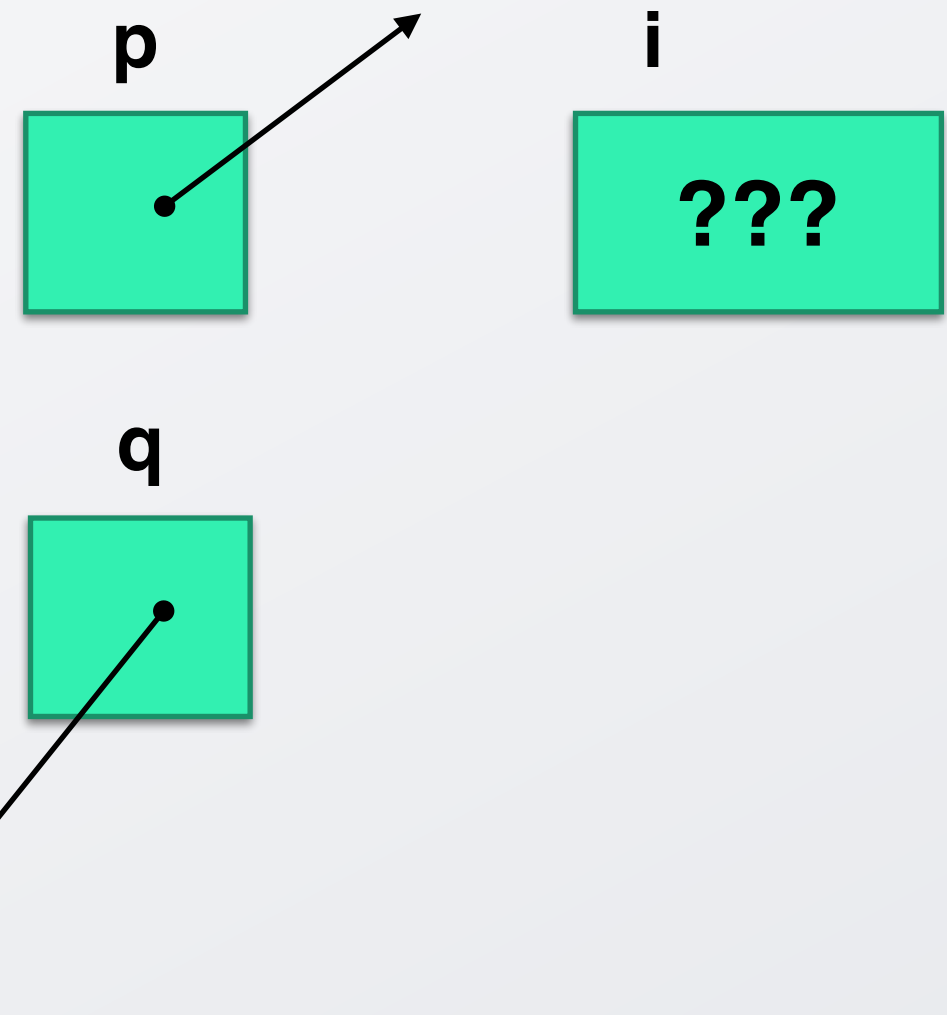
```
    p = &i;
```

```
    q = p;
```

```
    *p = 1;
```

```
    *q = 2;
```

```
}
```



```
void foo() {
```

```
    int i, *p, *q;
```

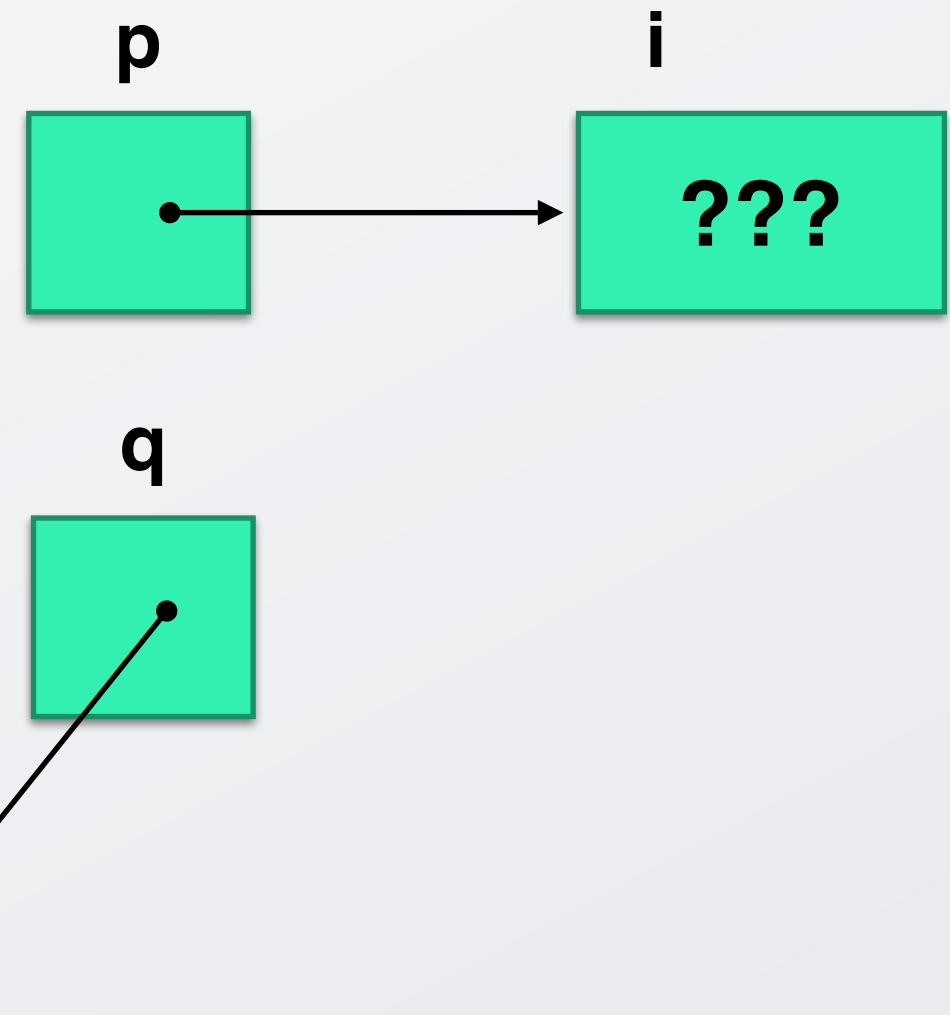
```
    p = &i;
```

```
    q = p;
```

```
    *p = 1;
```

```
    *q = 2;
```

```
}
```



```
void foo() {
```

```
    int i, *p, *q;
```

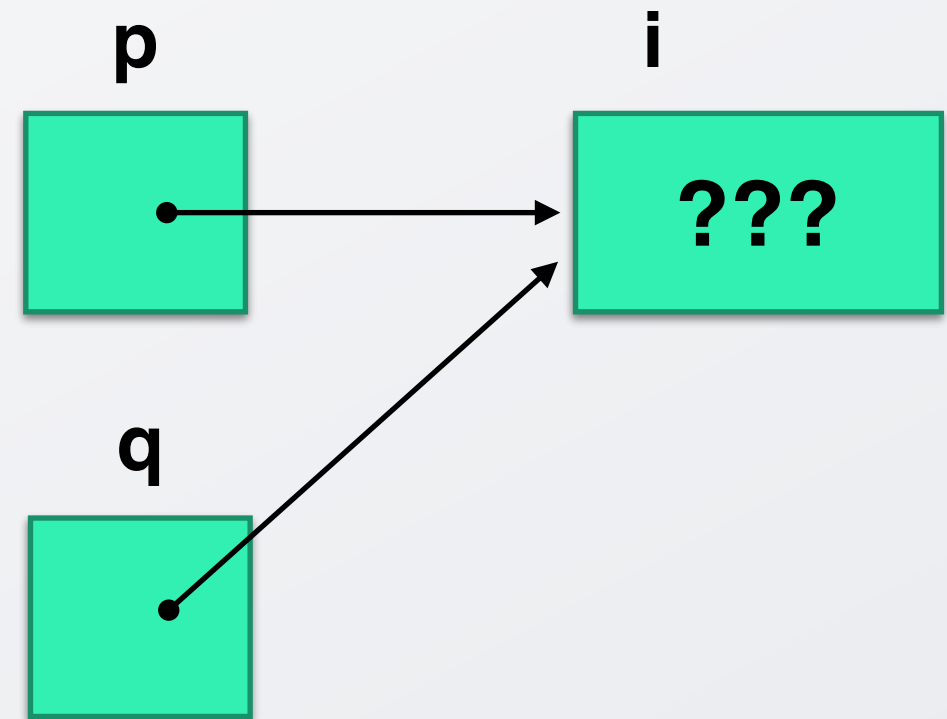
```
    p = &i;
```

```
    q = p;
```

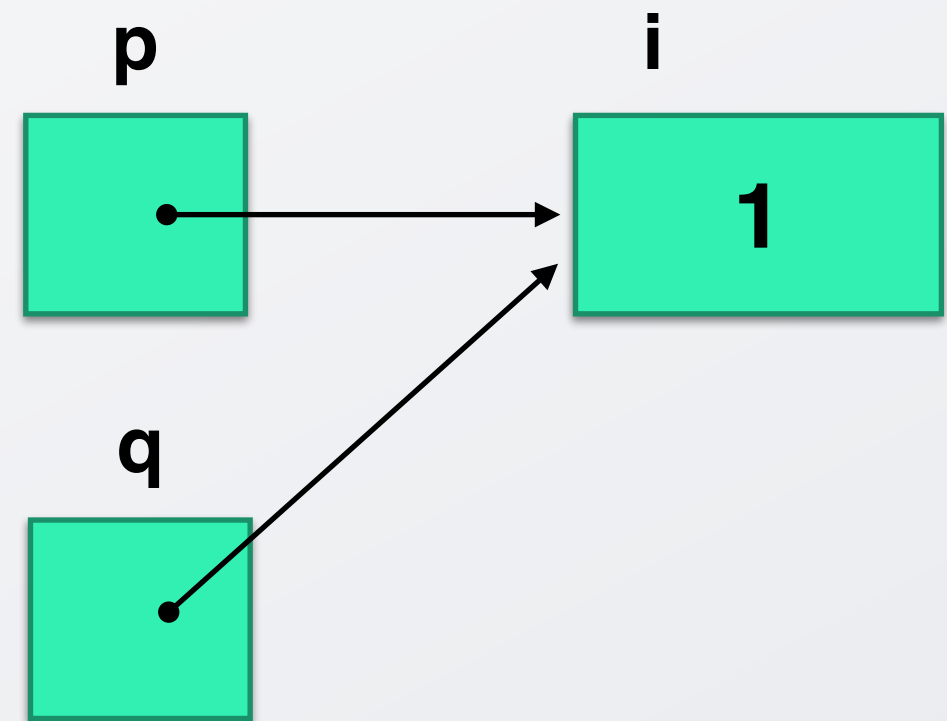
```
    *p = 1;
```

```
    *q = 2;
```

```
}
```



```
void foo() {  
    int i, *p, *q;  
  
    p = &i;  
    q = p;  
  
    *p = 1;  
  
    *q = 2;  
}
```



```
void foo() {
```

```
    int i, *p, *q;
```

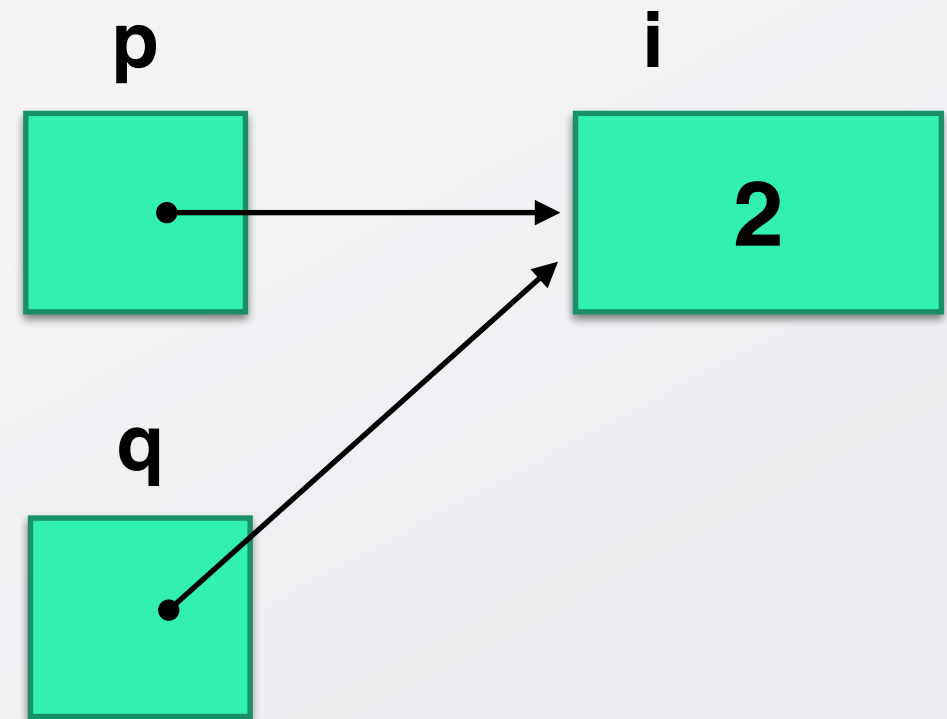
```
    p = &i;
```

```
    q = p;
```

```
    *p = 1;
```

```
    *q = 2;
```

```
}
```







交换两个整数变量

```
void
swap(int a, int b) // not work
{
    int temp = a;
    a = b;
    b = temp;
}
```

```
void
swap_int(int *pa, int *pb)
{
    int temp = *pa;
    *pa = *pb;
    *pb = temp;
}
```

```
int i = 10;
```

```
int j = 20;
```

```
swap(i, j);
```

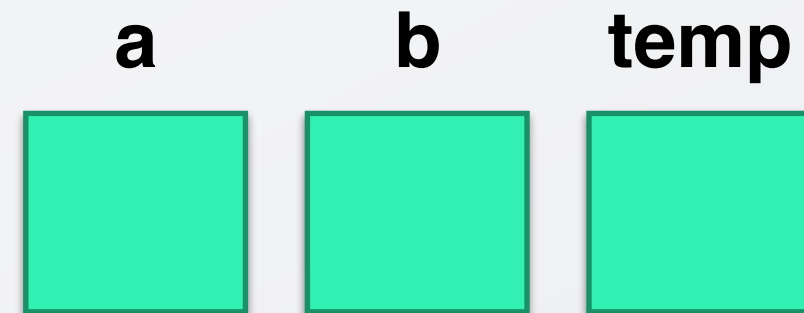
i

10

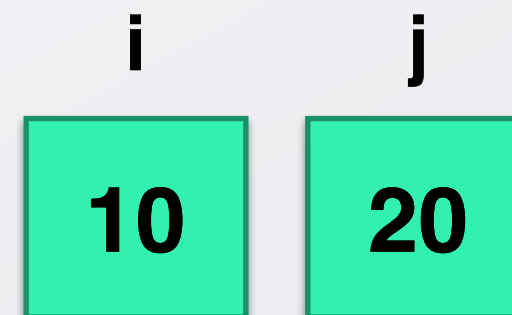
j

20

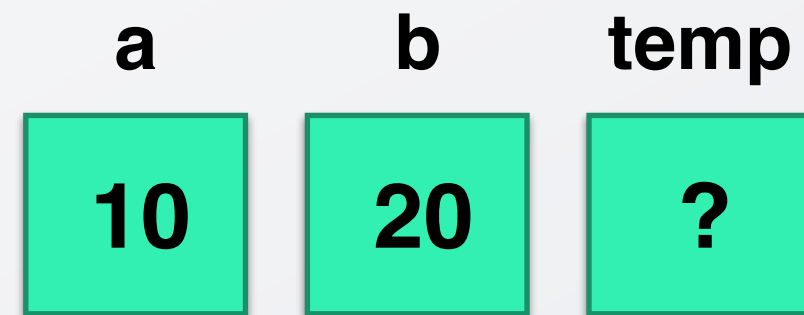
```
void  
swap(int a, int b) // not work  
{  
    int temp = a;  
    a = b;  
    b = temp;  
}
```



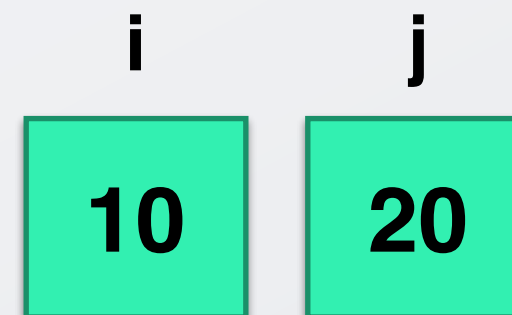
```
int i = 10;  
int j = 20;  
swap(i, j);
```



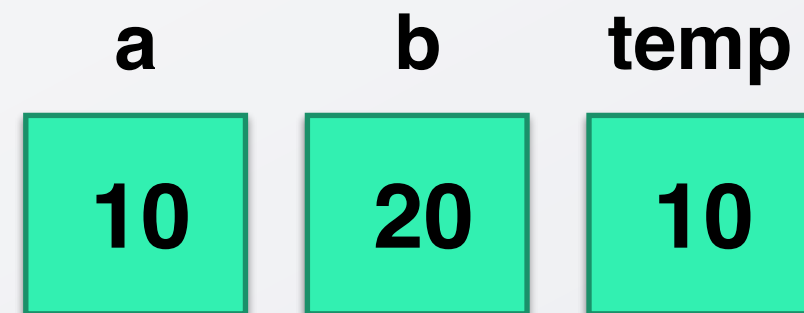
```
void  
swap(int a, int b) // not work  
{  
    int temp = a;  
    a = b;  
    b = temp;  
}
```



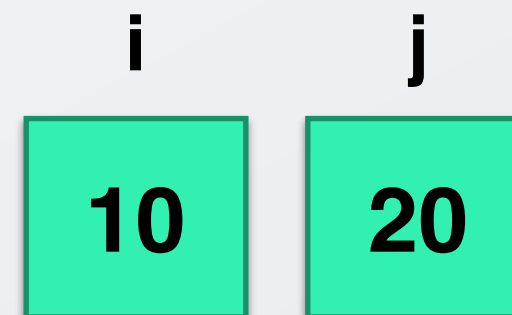
```
int i = 10;  
int j = 20;  
swap(i, j);
```



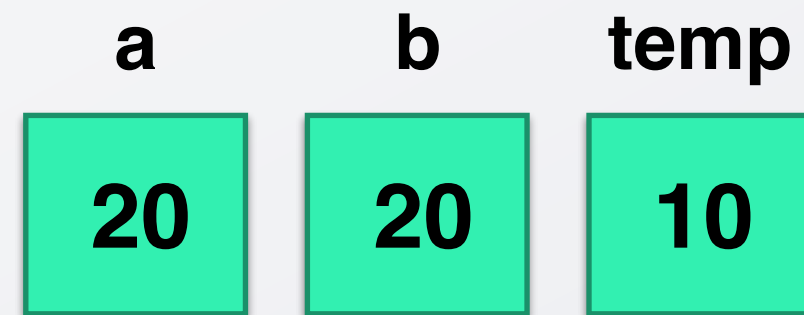
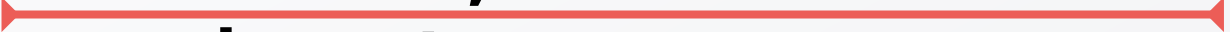
```
void  
swap(int a, int b) // not work  
{  
    int temp = a;  
    a = b;  
    b = temp;  
}
```



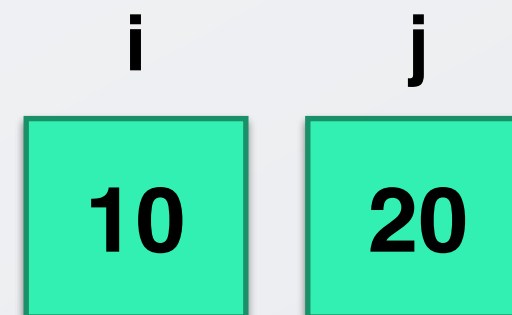
```
int i = 10;  
int j = 20;  
swap(i, j);
```



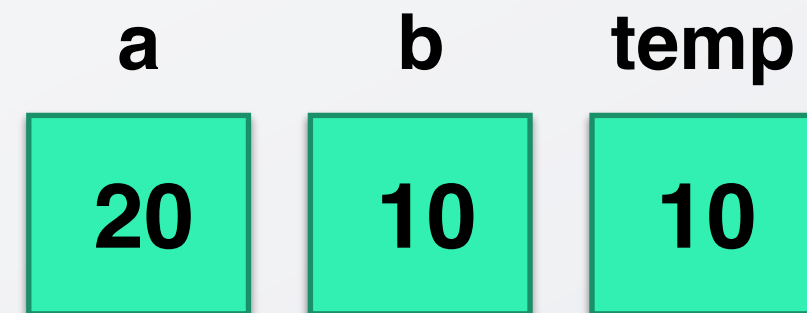

```
void
swap(int a, int b) // not work
{
    int temp = a;
    a = b;
    b = temp;
}
```



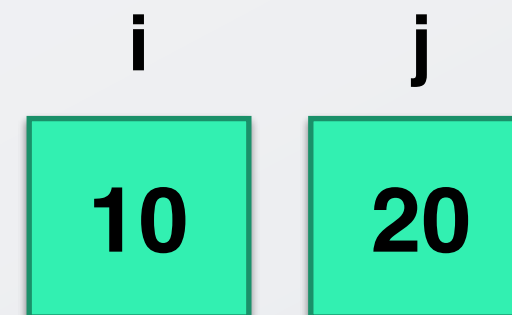
```
int i = 10;
int j = 20;
swap(i, j);
```



```
void  
swap(int a, int b) // not work  
{  
    int temp = a;  
    a = b;  
    b = temp;  
}
```

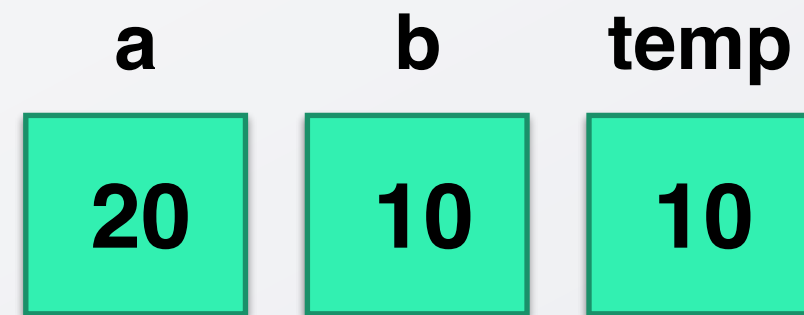
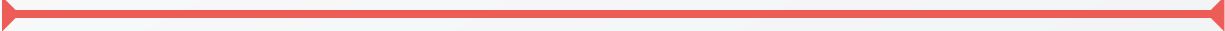


```
int i = 10;  
int j = 20;  
swap(i, j);
```

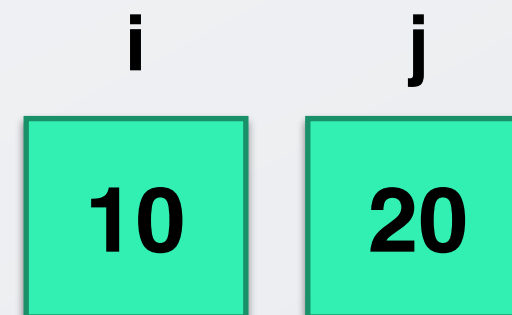




```
void  
swap(int a, int b) // not work  
{  
    int temp = a;  
    a = b;  
    b = temp;  
}
```



```
int i = 10;  
int j = 20;  
swap(i, j);
```



i, j 两个变量中的内容**并未交换**

```
int i = 10;  
int j = 20;  
swap(i, j);
```

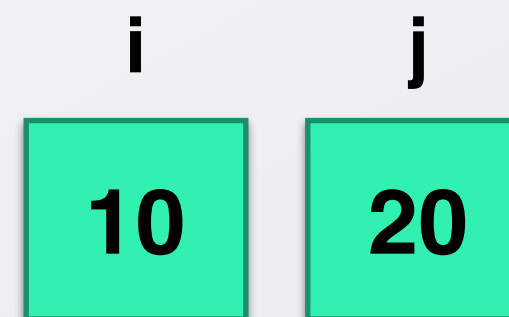




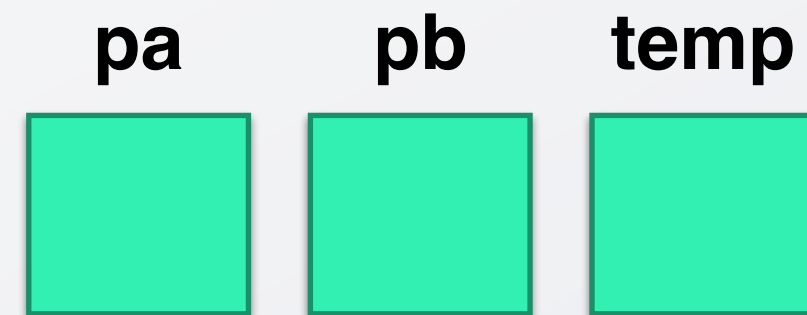
```
int i = 10;
```

```
int j = 20;
```

```
swap_int(&i, &j);
```



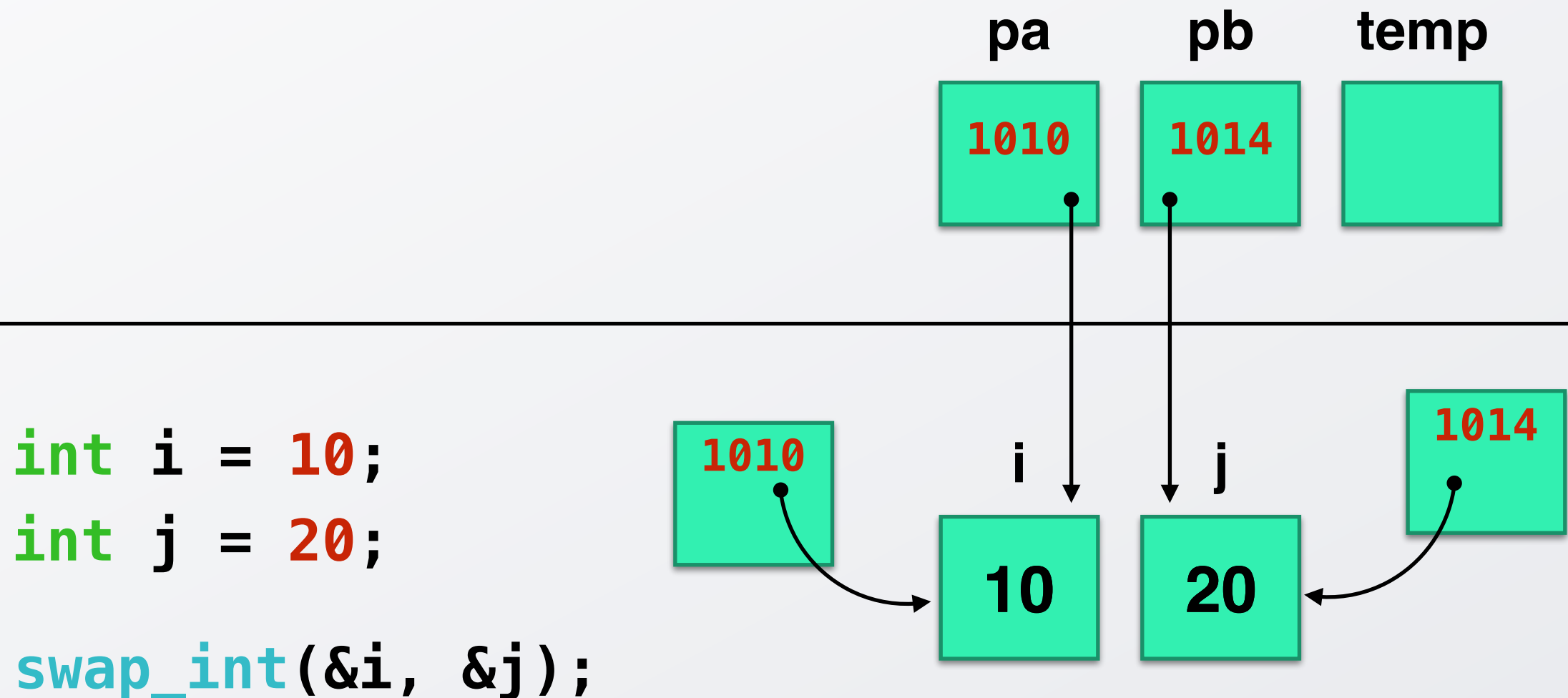
```
void  
swap_int(int *pa, int *pb)  
{  
    int temp = *pa;  
    *pa = *pb;  
    *pb = temp;  
}
```



```
int i = 10;  
int j = 20;  
swap_int(&i, &j);
```



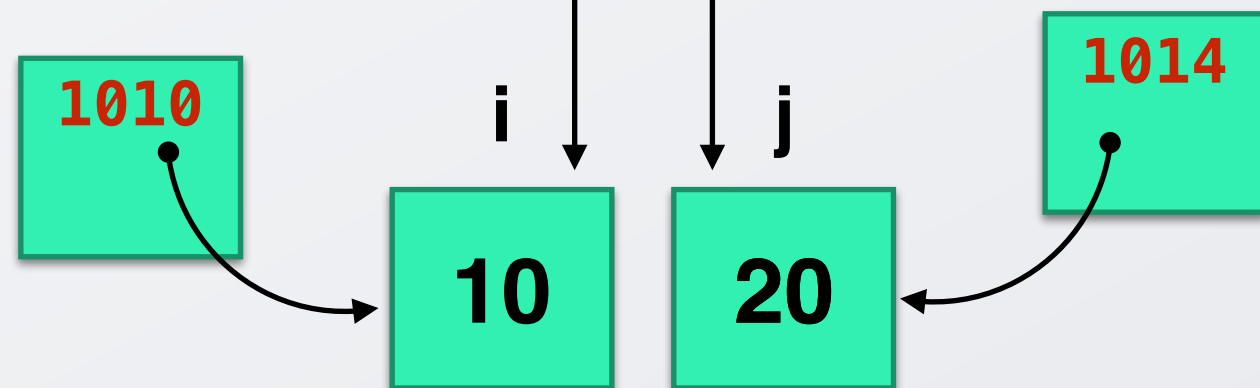
```
void  
swap_int(int *pa, int *pb)  
{  
    int temp = *pa;  
    *pa = *pb;  
    *pb = temp;  
}
```



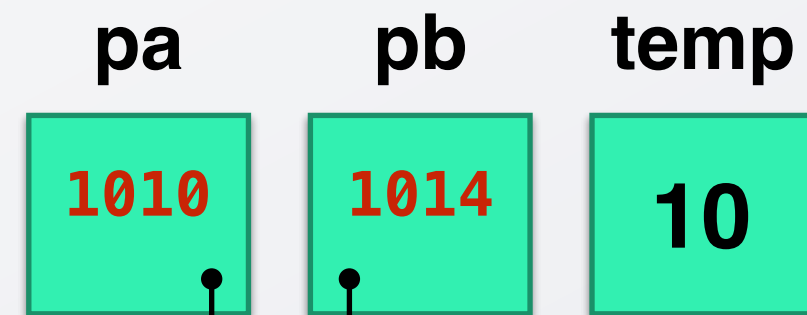
```
void  
swap_int(int *pa, int *pb)  
{  
    int temp = *pa;  
    *pa = *pb;  
    *pb = temp;  
}
```



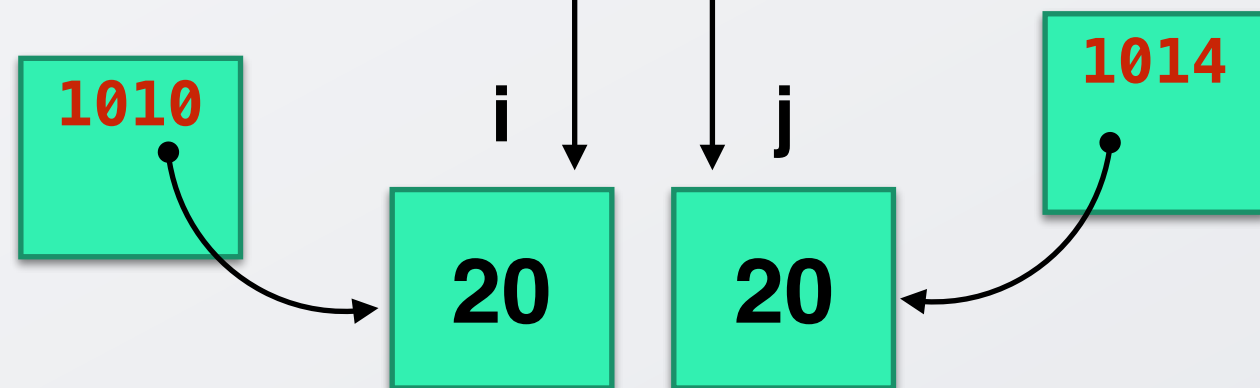
```
int i = 10;  
int j = 20;  
swap_int(&i, &j);
```



```
void  
swap_int(int *pa, int *pb)  
{  
    int temp = *pa;  
    *pa = *pb;  
    *pb = temp;  
}
```



```
int i = 10;  
int j = 20;  
swap_int(&i, &j);
```

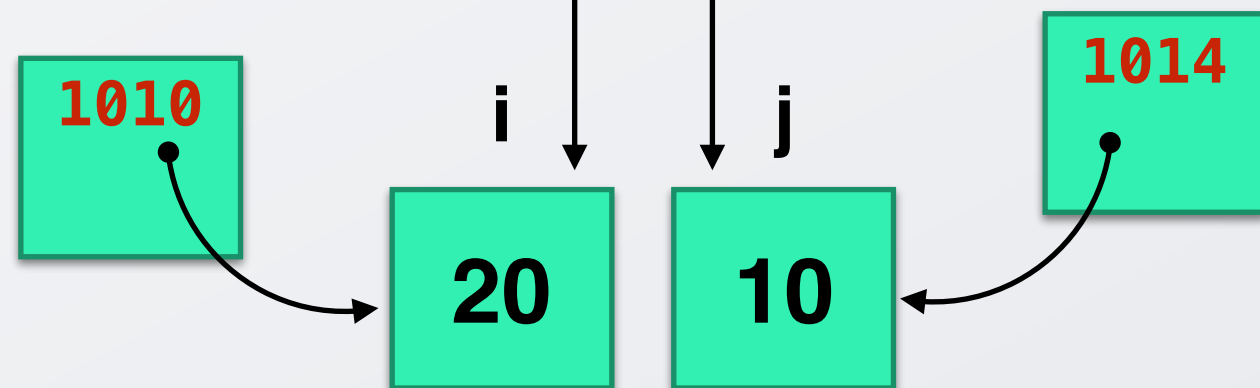




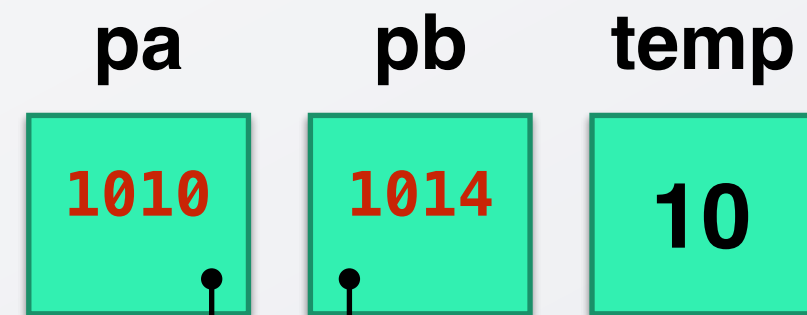
```
void  
swap_int(int *pa, int *pb)  
{  
    int temp = *pa;  
    *pa = *pb;  
    *pb = temp;  
}
```



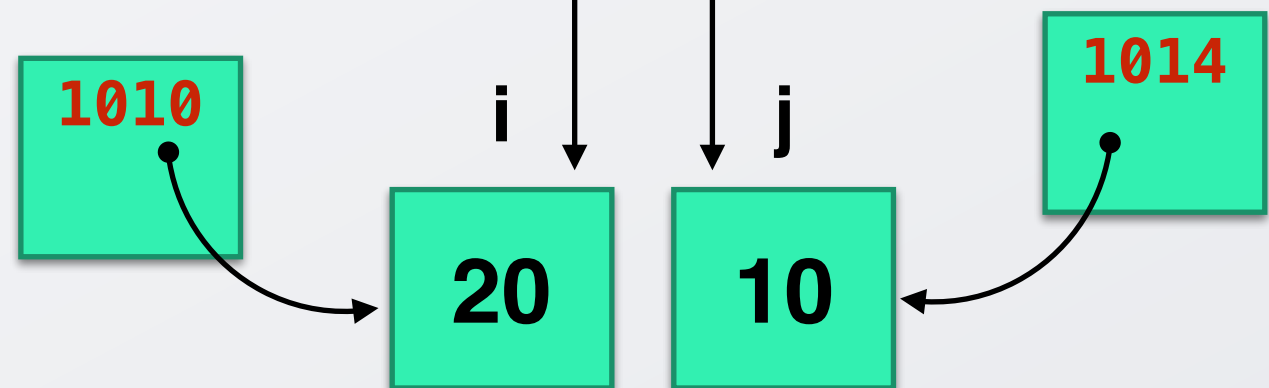
```
int i = 10;  
int j = 20;  
swap_int(&i, &j);
```



```
void  
swap_int(int *pa, int *pb)  
{  
    int temp = *pa;  
    *pa = *pb;  
    *pb = temp;  
}
```



```
int i = 10;  
int j = 20;  
swap_int(&i, &j);
```



```
int i = 10;
```

```
int j = 20;
```

```
swap_int(&i, &j);
```

i

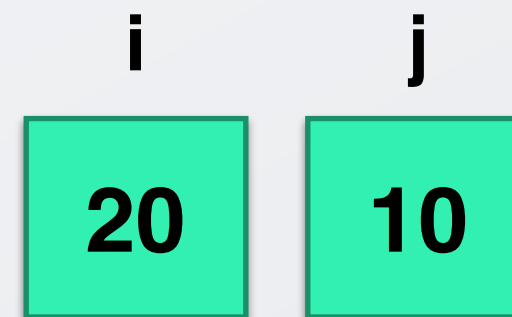
j

20

10

i, j 两个变量中的内容**交换成功**

```
int i = 10;  
int j = 20;  
swap_int(&i, &j);
```



# 图像处理初步



```
#include <stdio.h>

#define IMAGE_SIZE 3

typedef int image_t[IMAGE_SIZE][IMAGE_SIZE];

typedef image_t *ptr_image_t;

image_t image = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};

void swap(int *pa, int *pb) {
    int temp = *pa;
    *pa = *pb;
    *pb = temp;
}
```

```
ptr_image_t horiz_flip(ptr_image_t pimg) {  
    for (int row = 0; row < IMAGE_SIZE; row++) {  
        for (int col = 0; col < IMAGE_SIZE / 2; col++) {  
            swap(&(*pimg)[row][col],  
                &(*pimg)[row][IMAGE_SIZE - col - 1]);  
        }  
    }  
  
    return pimg;  
}
```



```
ptr_image_t transpose(ptr_image_t pimg) {  
    for (int row = 0; row < IMAGE_SIZE; row++) {  
        for (int col = row + 1; col < IMAGE_SIZE; col++) {  
            swap(&((*pimg)[row][col]), &((*pimg)[col][row]));  
        }  
    }  
  
    return pimg;  
}
```

```
ptr_image_t print_image(ptr_image_t pimg) {  
    for (int row = 0; row < IMAGE_SIZE; row++) {  
        for (int col = 0; col < IMAGE_SIZE; col++)  
            printf("%2d", (*pimg)[row][col]);  
  
        putchar( '\n' );  
    }  
    putchar( '\n' );  
    return pimg;  
}
```

```
ptr_image_t rotate_ccw(ptr_image_t pimg) {  
    return transpose(horiz_flip(pimg));  
}
```

```
ptr_image_t rotate(ptr_image_t pimg) {  
    return horiz_flip(transpose(pimg));  
}
```

```
ptr_image_t vert_flip(ptr_image_t pimg) {  
    return rotate_ccw(horiz_flip(rotate(pimg)));  
}
```

```
int main(int argc, char *argv[]) {  
  
    print_image(&image);  
    print_image(vert_flip(&image));  
    print_image(rotate(&image));  
    print_image(horiz_flip(&image));  
    print_image(transpose(&image));  
  
    return 0;  
}
```

# 抽象数据类型

## 函数 与 复合数据类型

func()

unsigned double  
char signed  
long short

### 存储操作

static extern  
int auto register  
= unsigned int

& ~ -(单目)

& \*

### 运算符与表达式

+ - \* / % ! || && == != < > <= >=

switch continue

### 程序流程控制

break while  
goto if...else... for

# 库函数

cos  
fabs

putchar

printf

strlen



WENZHENG COLLEGE OF SOOCHOW UNIVERSITY

2017.3.29







Soochow University

# 附录

