

广东白云学院 大数据协会

5： 指针

广东白云学院 大数据协会

5： 指针

如何在函数中交换两个数的值？

如何在函数中交换两个数的值?

```
#include<stdio.h>

void swap(int a, int b)
{
    int t = a;
    a = b;
    b = t;
}
```

Just For Fun

如何在函数中交换两个数的值?

```
#include<stdio.h>

void swap(int a, int b)
{
    printf("%d %d\n", a, b);
    a += b;
    b = a - b; // 只是为了好玩
    a = a - b; // 不要在你的代码里面这么写
    printf("%d %d\n", a, b);
}
```

- LeetCode 334: 不使用额外空间反转字符串

如何在函数中交换两个数的值？

```
#include<stdio.h>

void swap(int a, int b) {
    int temp;
    temp = a;
    a = b;
    b = temp;
}

int main(void) {
    int a = 3, b = 5;
    swap(a, b);
    printf("a = %d, b = %d\n", a, b);
    return 0;
}
```

- 输出结果会是什么？

如何在函数中交换两个数的值？

```
#include<stdio.h>

void swap(int a, int b) {
    int temp;
    temp = a;
    a = b;
    b = temp;
}

int main(void) {
    int a = 3, b = 5;
    swap(a, b);
    printf("a = %d, b = %d\n", a, b);
    return 0;
}
```

a = 3, b = 5

函数：不能直接修改参数

- 函数会复制**主函数**传入的参数
- 在函数中做出的修改都是对**拷贝**的修改
- 想要在函数中修改主函数参数，需要**指针**

指针

- 指针就是地址
- 直接修改主函数参数的地址的值，就能在函数中修改主函数参数

指针

```
#include<stdio.h>

void swap(int* a, int* b) {
    int temp;
    temp = *a;
    *a = *b;
    *b = temp;
}
int main(void) {
    int a = 3, b = 5;
    swap(&a, &b);
    printf("a = %d, b = %d\n", a, b);
    return 0;
}
```

a = 5, b = 3

指针

```
int* p; // 声明一个指向int类型的指针
*p = 8; // 使用 '*' 号取出地址的值

scanf("%d", &p); // 使用 '&' 号取指针p的地址

int a = 3, b = 5;
swap(&a, &b); // 不是指针也能用 '&' 取地址
```

Coding Time

LeetCode 344. 反转字符串

编写一个函数，其作用是将输入的字符串反转过来。输入字符串以字符数组 `s` 的形式给出。

```
void reverseString(char* s, int sSize);
```

不用额外空间反转字符串

```
void reverseString(char* s, int sSize) {  
    int left = 0;  
    int right = sSize - 1;  
    while (left < right) {  
        // 不是指针吗？怎么成数组了  
        s[left] += s[right];  
        s[right] = s[left] - s[right];  
        s[left] = s[left] - s[right];  
  
        left++;  
        right--;  
    }  
}
```

指针 = 数组

在计算机硬件中，没有指针和数组。只有**寻址**。

寻址：开始位置 + 偏移量。

指针：存储一个头节点地址，取 `p++` 时，增加一个偏移量
(例如，`int`类型增加4字节；`double`类型增加8字节)

数组：存储 `arr[0]` 的地址，取 `arr[i]` 时，加上
`i*sizeof(type)`

拓展内容

PA1 排序算法

广东白云学院 大数据协会

感谢聆听！