**1.递归函数解决逆序输出字符串**

void reverse(char\* str)

{

char tmp = \*str;

int len = strlen(str); //1

\*str = \*(str + len - 1); //2

\*(str + len - 1) = '\0'; //3

if(len>1) //老师这里的条件是strlen(str+1)>=2

reverse(str + 1); //4

\*(str+len-1)= tmp;

}

**2.交换两数但不引入新的变量：**

a=a+b

b=a-b; 可能会溢出

a=a-b;

a=a^b;

b=a^b;

a=a^b;

1. **求一个整数存储在内存中的二进制里1的个数**

#include <stdio.h>

循环移位

int count\_bits\_method1(int num) {

int count = 0;

int bits = sizeof(num) \* 8; // 整数的总位数

for (int i = 0; i < bits; i++) {

if (num & 1) { // 检查最低位是否为1 未考虑负整数，且低效

count++;

}

num >>= 1; // 右移一位

}

return count;

}

高效：位运算符

int count\_bits\_method3(int num) {

int count = 0;

unsigned int unum = (unsigned int)num;

while (unum) {

count++;

unum = unum & (unum - 1); // 每次消除最低位的1

}

return count;

}

%2，/2：和10进制数里求每位的数字一样，但是记得考虑负数，这时候传参应该用无符号整数来计算

去除二进制数的某一位 (num>>i)&1;