

C语言头文件常用函数

#include <stdio.h>

- scanf();
 - while(scanf("%d",&num)!=EOF){ }
 - 遇到空格、换行、EOF则不会收
- fgets (char buffer[],n, stdin)
 - 遇到空格、换行也会接收，英语单词会用到这个函数
 - n是指最大可以接收的字符，包括结尾的'\0',所以实际可输入的字符是n-1个。
- printf();
- getchar(void);
 - int num=0;while(num=getchar())&&num!='\0'|| num!=EOF), 这段代码的作用是清除缓冲区里面的字符，遇到EOF和'\n'时才会停止。
- sprintf(char* buffer,"%d",int num);
 - 将整数转化为字符串；
 - 第二个参数可以动态更改，%d 有符号十进制整数，%u 无符号十进制整数，%x 无符号十六进制整数（小写），%X 无符号十六进制整数（大写），%o 无符号八进制整数
- snprintf(char str[], sizeof(str)缓冲池大小, "%s%d....", 某个字符串, 某个整数);
 - 返回一个整数，这个整数表示往buffer里面写了多少数据，不包括'\0'，如果要写入的数据超过了buffer的容量，那么只返回数据的长度，不往buffer写入，所以可以由此判断是否写入成功
 - 第三个参数，可以灵活一些，"%d你好啊%s"

#include <math.h>

- pow(num1,num2); num1的num2次方；
- sqrt(num1); 开平方
- ceil(num1); 向上取整
- floor(num1); 向下取整

#include <stdlib.h>

- malloc(sizeof(XXX))
- free();
- calloc(sizeof(XXX),num); 每个多大，多少个
- realloc(原来的空间指针，新的空间大小) 记得强制类型转换
- memcpy();
- atoi(char* str) 返回int
 - 将字符串转化为整数；注意 atoi 函数在处理非数字字符串时不会报错，而是返回0。如果输入可能包含非法字符，建议使用更安全的函数（如 strtol）。
- qsort(void* arr,排序元素个数, sizeof(数组每个元素多大), int (*fcmp)(const void *, const void *))
 - int (*fcmp)(const void *, const void *)是一个指向比较函数的指针。比较函数用于定义排序的规则。它接收两个参数（通常是指向数组中两个元素的指针），并返回一个整数值：
 - 如果第一个参数小于第二个参数，返回负值。
 - 如果两个参数相等，返回 0。
 - 如果第一个参数大于第二个参数，返回正值。

#include<string.h>

- strcmp(str1,str2); 返回0表示相等
- strncmp(str1,str2,num) 比较前n个字符
- strncat(dest,src,num);
 - strcat 依赖于目标字符串的终止符 \0 来确定追加的起始位置。如果目标字符串没有正确以 \0 结尾，strcat 会从错误的位置开始追加，导致未定义行为。
 - 将src拼到dest上
- strcpy(dest, src);
 - 将src的内容复制到dest上
 - 确保dest空间能够装下
- strncpy (dest, src, n)
 - strncpy 不会自动添加 \0，要预留一个位置手动加'\0'
- strchr(char* str,char c)
 - strchr 是 C 标准库中的一个函数，用于在字符串中查找指定字符的第一次出现位置。它返回指向该字符的指针，如果未找到该字符，则返回 NULL。
 - 查找的首次出现的字符
 - 返回指向该字符的指针
- strrchr(char* str,char c)
 - 查找的最后出现的字符
 - 返回指向该字符的指针
- strstr(char* father,char *son)
 - 用于在father字符串中查找son字符串的第一次出现位置,返回指向该子字符串的指针,如果未找到子字符串，则返回 NULL
 - strstr 是区分大小写的
- strtok(char str[],char* target)
 - char src[]="abc,de.fgh-igk,l,mno.pq.rst-uv,wx.yz-";
char* target=",-.";
 - char* result=strtok(src,target);
while(result!=NULL){
printf("%s\n",result);
result=strtok(NULL,target);
}这么用，记住循环中的NULL
 - 会返回一个char*，每次字符串被一分为二，返回的char*指向前一半
 - 通过target间隔符将str分开，要确保str是可读的
- strspn (char* str, char* ma)
 - 看str开头的有多少位在ma中
 - 返回一个无符号整数