

一、数学函数

调用数学函数时，要求在源文件中包下以下命令行：

```
#include <math.h>
```

函数原型说明	功能	返回值	说明
int abs(int x)	求整数x的绝对值	计算结果	
double fabs(double x)	求双精度实数x的绝对值	计算结果	
double acos(double x)	计算 $\cos^{-1}(x)$ 的值	计算结果	x在-1 ~ 1范围内
double asin(double x)	计算 $\sin^{-1}(x)$ 的值	计算结果	x在-1 ~ 1范围内
double atan(double x)	计算 $\tan^{-1}(x)$ 的值	计算结果	
double atan2(double x)	计算 $\tan^{-1}(x/y)$ 的值	计算结果	
double cos(double x)	计算 $\cos(x)$ 的值	计算结果	x的单位为弧度
double cosh(double x)	计算双曲余弦 $\cosh(x)$ 的值	计算结果	
double exp(double x)	求 e^x 的值	计算结果	
double fabs(double x)	求双精度实数x的绝对值	计算结果	
double floor(double x)	求不大于双精度实数x的最大整数		
double fmod(double x,double y)	求x/y整除后的双精度余数		
double frexp(double val,int *exp)	把双精度val分解尾数和以2为底的指数n，即 $val=x*2^n$ ，n存放在exp所指的变量中	返回位数x $0.5 \leq x < 1$	
double log(double x)	求 $\ln x$	计算结果	$x > 0$
double log10(double x)	求 $\log_{10}x$	计算结果	$x > 0$
double modf(double val,double *ip)	把双精度val分解成整数部分和小数部分，整数部分存放在ip所指的变量中	返回小数部分	
double pow(double x,double y)	计算 x^y 的值	计算结果	
double sin(double x)	计算 $\sin(x)$ 的值	计算结果	x的单位为弧度
double sinh(double x)	计算x的双曲正弦函数 $\sinh(x)$ 的值	计算结果	
double sqrt(double x)	计算x的开方	计算结果	$x \geq 0$
double tan(double x)	计算 $\tan(x)$	计算结果	
double tanh(double x)	计算x的双曲正切函数 $\tanh(x)$ 的值	计算结果	

二、字符函数

调用字符函数时，要求在源文件中包下以下命令行：

```
#include <ctype.h>
```

函数原型说明	功能	返回值
int isalnum(int ch)	检查ch是否为字母或数字	是，返回1；否则返回0
int isalpha(int ch)	检查ch是否为字母	是，返回1；否则返回0
int iscntrl(int ch)	检查ch是否为控制字符	是，返回1；否则返回0
int isdigit(int ch)	检查ch是否为数字	是，返回1；否则返回0
int isgraph(int ch)	检查ch是否为ASCII码值在ox21到ox7e的可打印字符（即不包含空格字符）	是，返回1；否则返回0
int islower(int ch)	检查ch是否为小写字母	是，返回1；否则返回0
int isprint(int ch)	检查ch是否为包含空格符在内的可打印字符	是，返回1；否则返回0
int ispunct(int ch)	检查ch是否为除了空格、字母、数字之外的可打印字符	是，返回1；否则返回0
int isspace(int ch)	检查ch是否为空格、制表或换行符	是，返回1；否则返回0
int isupper(int ch)	检查ch是否为大写字母	是，返回1；否则返回0

int isxdigit(int ch)	检查ch是否为16进制数	是，返回1；否则返回0
int tolower(int ch)	把ch中的字母转换成小写字母	返回对应的小写字母
int toupper(int ch)	把ch中的字母转换成大写字母	返回对应的大写字母

三、字符串函数

调用字符函数时，要求在源文件中包下以下命令行：

```
#include <string.h>
```

函数原型说明	功能	返回值
char *strcat(char *s1,char *s2)	把字符串s2接到s1后面	s1所指地址
char *strchr(char *s,int ch)	在s所指字符串中，找出第一次出现字符ch的位置	返回找到的字符的地址，找不到返回NULL
int strcmp(char *s1,char *s2)	对s1和s2所指字符串进行比较	s1<s2,返回负数； s1= =s2,返回0； s1>s2,返回正数
char *strcpy(char *s1,char *s2)	把s2指向的串复制到s1指向的空间	s1 所指地址
unsigned strlen(char *s)	求字符串s的长度	返回串中字符（不计最后的'\0'）个数
char *strstr(char *s1,char *s2)	在s1所指字符串中，找出字符串s2第一次出现的位置	返回找到的字符串的地址，找不到返回NULL

四、输入输出函数

调用字符函数时，要求在源文件中包下以下命令行：

#include <stdio.h>

函数原型说明	功能	返回值
void clearer(FILE *fp)	清除与文件指针fp有关的所有出错信息	无
int fclose(FILE *fp)	关闭fp所指的文件，释放文件缓冲区	出错返回非0，否则返回0
int feof (FILE *fp)	检查文件是否结束	遇文件结束返回非0，否则返回0
int fgetc (FILE *fp)	从fp所指的文件中取得下一个字符	出错返回EOF，否则返回所读字符
char *fgets(char *buf,int n, FILE *fp)	从fp所指的文件中读取一个长度为n-1的字符串，将其存入buf所指存储区	返回buf所指地址，若遇文件结束或出错返回NULL
FILE *fopen(char *filename,char *mode)	以mode指定的方式打开名为filename的文件	成功，返回文件指针（文件信息区的起始地址），否则返回NULL
int fprintf(FILE *fp, char *format, args,...)	把args,...的值以format指定的格式输出到fp指定的文件中	实际输出的字符数
int fputc(char ch, FILE *fp)	把ch中字符输出到fp指定的文件中	成功返回该字符，否则返回EOF
int fputs(char *str, FILE *fp)	把str所指字符串输出到fp所指文件	成功返回非负整数，否则返回-1（EOF）
int fread(char *pt,unsigned size,unsigned n, FILE *fp)	从fp所指文件中读取长度size为n个数据项存到pt所指文件	读取的数据项个数
int fscanf (FILE *fp, char *format,args,...)	从fp所指的文件中按format指定的格式把输入数据存入到args,...所指的内存中	已输入的数据个数，遇文件结束或出错返回0
int fseek (FILE *fp,long offer,int base)	移动fp所指文件的位置指针	成功返回当前位置，否则返回非0
long ftell (FILE *fp)	求出fp所指文件当前的读写位置	读写位置，出错返回 -1L
int fwrite(char *pt,unsigned size,unsigned n, FILE *fp)	把pt所指指向的n*size个字节输入到fp所指文件	输出的数据项个数
int getc (FILE *fp)	从fp所指文件中读取一个字符	返回所读字符，若出错或文件结束返回EOF
int getchar(void)	从标准输入设备读取下一个字符	返回所读字符，若出错或文件结束返回-1
char *gets(char *s)	从标准设备读取一行字符串放入s所指存储区，用'\0'替换读入的换行符	返回s,出错返回NULL
int printf(char *format,args,...)	把args,...的值以format指定的格式输出到标准输出设备	输出字符的个数
int putc (int ch, FILE *fp)	同fputc	同fputc
int putchar(char ch)	把ch输出到标准输出设备	返回输出的字符，若出错则返回EOF
int puts(char *str)	把str所指字符串输出到标准设备，将'\0'转成回车换行符	返回换行符，若出错，返回EOF
int rename(char *oldname,char *newname)	把oldname所指文件名改为newname所指文件名	成功返回0，出错返回-1
void rewind(FILE *fp)	将文件位置指针置于文件开头	无
int scanf(char *format,args,...)	从标准输入设备按format指定的格式把输入数据存入到args,...所指的内存中	已输入的数据的个数

五、动态分配函数和随机函数

调用字符函数时，要求在源文件中包下以下命令行：

```
#include <stdlib.h>
```

函数原型说明	功能	返回值
void *calloc(unsigned n,unsigned size)	分配n个数据项的内存空间，每个数据项的大小为size个字节	分配内存单元的起始地址；如不成功，返回0
void *free(void *p)	释放p所指的内存区	无
void *malloc(unsigned size)	分配size个字节的存储空间	分配内存空间的地址；如不成功，返回0
void *realloc(void *p,unsigned size)	把p所指内存区的大小改为size个字节	新分配内存空间的地址；如不成功，返回0
int rand(void)	产生0 ~ 32767的随机整数	返回一个随机整数
void exit(int state)	程序终止执行，返回调用过程，state为0正常终止，非0非正常终止	无