

语言操作函数

stdio.h

`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 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

`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

math.h

`int abs(int x)`

`double fabs(double x)`

`double exp(double x)` : 返回e的x次方

`double pow(double x,double y)`

`double sqrt(double x)`

`double ceil (double);` 取上整，返回不比x小的最小整数

`double floor (double);` 取下整，返回不比x大的最大整数，即 高斯函数[x]

ctype.h

`int isdigit(int ch)` 检查ch是否为数字 是，返回1；否则返回0

`int islower(int ch)`

`int isupper(int ch)`

`int tolower(int ch)` ch改为小写

`int toupper(int ch)` ch改为大写

string.h

`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小,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

stdlib.h

`void *free(void *p)` 释放p所指的内存区

`void *malloc(unsigned size)` 分配size个字节的存储空间;如不成功，返回0

limits.h

库宏

宏	值	描述
CHAR_BIT	8	定义一个字节的比特数。
SCHAR_MIN	-128	定义一个有符号字符的最小值。
SCHAR_MAX	127	定义一个有符号字符的最大值。
UCHAR_MAX	255	定义一个无符号字符的最大值。
CHAR_MIN	0	定义类型 char 的最小值，如果 char 表示负值，则它的值等于 SCHAR_MIN，否则等于 0。
CHAR_MAX	127	定义类型 char 的最大值，如果 char 表示负值，则它的值等于 SCHAR_MAX，否则等于 UCHAR_MAX。
MB_LEN_MAX	1	定义多字节字符中的最大字节数。
SHRT_MIN	-32768	定义一个短整型的最小值。
SHRT_MAX	+32767	定义一个短整型的最大值。
USHRT_MAX	65535	定义一个无符号短整型的最大值。
INT_MIN	-2147483648	定义一个整型的最小值。
INT_MAX	2147483647	定义一个整型的最大值。
UINT_MAX	4294967296	定义一个无符号整型的最大值。
LONG_MIN	-9223372036854775808	定义一个长整型的最小值。
LONG_MAX	9223372036854775807	定义一个长整型的最大值。
ULONG_MAX	1.8446744e+19	定义一个无符号长整型的最大值。

其它

排序方法	平均时间	最好时间	最坏时间
桶排序(不稳定)	$O(n)$	$O(n)$	$O(n)$
基数排序(稳定)	$O(n)$	$O(n)$	$O(n)$
归并排序(稳定)	$O(n\log n)$	$O(n\log n)$	$O(n\log n)$
快速排序(不稳定)	$O(n\log n)$	$O(n\log n)$	$O(n^2)$
堆排序(不稳定)	$O(n\log n)$	$O(n\log n)$	$O(n\log n)$
希尔排序(不稳定)	$O(n^{1.25})$		
冒泡排序(稳定)	$O(n^2)$	$O(n)$	$O(n^2)$
选择排序(不稳定)	$O(n^2)$	$O(n^2)$	$O(n^2)$
直接插入排序(稳定)	$O(n^2)$	$O(n)$	$O(n^2)$