

F C语言常用库函数整理

③C 库函数

F C语言常用库函数整理

1.stdio.h

2.stdlib.h

①分配空间

②算法相关

③其他

④库宏

3.string.h

4.math.h

①普通数字处理

②三角函数

③其他

5 ctype.h

附录

PS: `include` 称为文件包含命令, 即它把 `< >` 或 `" "` 内指定的文件包含到程序中。被包含的(头)文件通常是由系统提供的, 其扩展名为 `.h`。

如下介绍了stdio.h、stdlib.h、string.h、math.h等系统提供的文件。

1.stdio.h

标准 I/O 库。

- `int scanf(char *format, args, ...)`: 按 `format` 指定的格式将输入数据存入到 `args, ...` 指定的内存中(**读取到空格就停止**)。
- `int printf(char *format, args, ...)`: 将 `args, ...` 的值以 `format` 指定的格式输出。
- `char *gets(char *str)`: 从标准输入中读取一行, 并把它存储在 `str` 所指向的字符串中(**读取到换行符或者到达文件末尾时停止, 读取到空格不停止**)。
- `int getchar(void)`: 从标准输入中读取下一个字符。
| 如果没有错误发生, 返回相同的字符, 如果碰到文件结尾或者有错误发生, 则返回 EOF。
- `int putchar(int character)`: 将字符输出到标准输出中。
| 如果没有错误发生, 返回相同的字符, 如果有错误发生, 则返回 EOF。

- `int sprintf(char *str, const char *format, ...)`: 格式化输出到 `str` 所指向的字符串。

```
int a = 123;
char* s1 = "abc";
char* s2;
sprintf(s2, "%d %s", a, s1);
printf("%s", s2); // 123 abc。
```

- 宏 `EOF` 是一个负整数，通常在有函数调用错误或者碰到文件结尾时返回。

```
// 判断是否到达输入流末尾。
if (scanf("%d", %num) != EOF) {
    ...
}
```

2.stdlib.h

①分配空间

- `void *calloc(unsigned n, unsigned size)`: 分配所需的内存空间，并返回一个指向它的指针。
- `void *malloc(unsigned size)`: 分配所需的内存空间，并返回一个指向它的指针。

```
// 如：向系统申请分配100个int型的连续内存空间。
// 并返回所分配的内存空间的首地址。
int* array = malloc(sizeof(int)*100);
int* array = calloc(100, sizeof(int));
```

- `void *realloc(void *ptr, unsigned size)`: 重新调整之前调用 `malloc` 或 `calloc` 所分配的内存块的大小(其中 `ptr` 为指向内存块的指针)。
- `void free(void *ptr)`: 释放调用 `calloc`、`malloc` 或 `realloc` 所分配的内存空间(即释放 `ptr` 指针所指向的内存空间)。

②算法相关

- 执行快速排序:

```
// 主函数外声明一个compare函数，决定升序/降序。
int compare(const void* a, const void* b) {
    // 升序。
    return (*(待排元素类型*)a) -/> (*(待排元素类型*)b);
    // 降序。
    return (*(待排元素类型*)b) -/> (*(待排元素类型*)a);
}
```

```
}
```

// 主函数内调用快排函数，进行快速排序。

```
qsort(待排数组首地址, 待排元素总个数, sizeof(待排元素类型), compare(函数, 决定升序/降序));
```

• 执行二分查找：

// 主函数外声明一个compare函数，决定升序/降序。

```
int compare(const void* a, const void* b) {
    // 升序。
    return (*(待排元素类型*)a) -/> (*(待排元素类型*)b);
    // 降序。
    return (*(待排元素类型*)b) -/> (*(待排元素类型*)a);
}
```

// 主函数内调用二分查找函数，查找某个元素。

```
int* ptr;
ptr = (int*)bsearch(指向要查找元素的指针, 待查数组首地址, 待查数组的元素总个数, sizeof(待排元素类型), compare(函数, 决定升序/降序));
```

如果查找成功，则返回一个指向数组中匹配元素的指针，否则返回空指针。

③其他

- **int rand(void): 返回一个范围在 0 - RAND_MAX(32767) 之间的伪随机数。**

```
// 返回[0-10)之间的随机数。
printf("%d", rand()%10);
// 返回[20-30)之间的随机数。
printf("%d", rand()%10+20);
// 返回[m,n)范围内的随机数num, 可用:
printf("%d", rand()%(n-m)+m);
```

- **int abs(int x): 求整数 x 的绝对值。**
- **long int labs(long int x): 返回长整型 x 的绝对值。**
- **int atoi(const char *str): 把参数 str 所指向的字符串转换为一个 int 型整数。**

```
char* s = "521 abc";
int a = atoi(s);
printf("%d", a); // 521。
```

- **double atof(const char *str): 把参数 str 所指向的字符串转换为一个 double 型浮点数。**
- **long int atol(const char *str): 把参数 str 所指向的字符串转换为一个 long int 型长整数。**
- **void exit(int status): 使一个程序正常终止(status 为 0)，或非正常终止(status 为 1)。**
- **void abort(void): 使一个异常程序终止。**

④库宏

RAND_MAX: rand 函数返回的最大值。

NULL: 是一个空指针常量的值。

NULL—— `stdio.h`、`string.h` 头文件中也有此宏。

3.string.h

`string.h` 头文件定义了各种操作字符数组的函数。

- `int strcmp(const char *str1, const char *str2)`: 把 `str1` 所指向的字符串和 `str2` 所指向的字符串进行比较。

```
/*
    strcmp() 函数会按照ASCII码顺序比较两个字符串，并由函数返回值返回比较结果。
    字符串1 = 字符串2，返回值 = 0。
    字符串1 > 字符串2，返回值 > 0。
    字符串1 < 字符串2，返回值 < 0。
*/
char* s1 = "asd";
char* s2 = "asd";
// 两字符串相等，返回1，否则，返回0。
if (strcmp(s1, s2) == 0)
    return 1;
else
    return 0;
```

- `size_t strlen(const char *str)`: 计算字符串 `str` 的长度。
- `char *strcpy(char *dest, const char *src)`: 把 `src` 所指向的字符串复制到 `dest`。
- `char *strcat(char *dest, const char *src)`: 把 `src` 所指向的字符串追加到 `dest` 所指向的字符串的结尾。

```
// 注意：以下代码会出现错误。
char* s = malloc(sizeof(char)*10);
char arr[5] = "abcde";
strcat(s, arr); // x, 必须给s赋初值，再使用此函数。
```

```
// 正确操作。
char s[5] = "aaaaa";
char arr[5] = "bbbbb";
strcat(s, arr);
printf("%s", s); // aaaaabbbbb。
```

- `void *memcpy(void *str1, const void *str2, size_t n)`: 从存储区 `str2` 复制 `n` 个字符到存储区 `str1`。
| 返回一个指向目标存储区 `str1` 的指针。
- `char *strtok(char *str, const char *delim)`: 分解字符串 `str`, 其中 `delim` 为分隔符。
| `str` — 要被分解的字符串。
| `delim` — 包含分隔符的 C 字符串。

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

int main() {
    char s[15] = "I am so cold";
    char** arr = (char**)malloc(sizeof(char*)*4);
    int i = 0;

    arr[i] = (char*)malloc(sizeof(char)*5);
    arr[i] = strtok(s, " ");

    /*
    输出:
    I
    am
    so
    cold
    */
    while (arr[i] != NULL) {
        printf("%s\n", arr[i]);
        i++;
        arr[i] = (char*)malloc(sizeof(char)*5);
        arr[i] = strtok(NULL, " ");
    }

    return 0;
}
```

4.math.h

`math.h` 头文件定义了各种数学函数, 此库中所有函数都带有一个 `double` 类型的参数, 且都返回 `double` 类型的结果。

①普通数字处理

- `double fabs(double x)`: 求浮点数 `x` 的绝对值。

- **double sqrt(double x): 计算 x 的平方根。**
- **double pow(double x, double y): 计算 x 的 y 次幂。**
- double fmod (double x, double y): 返回 x / y 的余数。
- double exp(double x): 求 e 的 x 次幂, e = 2.718281828。
- double log(double x): 计算 x 的自然对数, 即 e 为底数, x 的值应大于零。
- double log10(double x): 计算 x 的常用对数, 即 10 为底数, x 的值应大于零。
- double floor(double x): 求不大于 x 的最大整数。
| 返回 x 的下限, 如 74.12 的下限为 74, -74.12 的下限为 -75, 返回值为 double 类型。
- double ceil(double x): 求不小于 x 的最小整数。
| 返回 x 的上限, 如 74.12 的上限为 75, -74.12 的上限为 -74, 返回值为 double 类型。

②三角函数

- double cos(double x): 求 x (弧度表示)的余弦值。
- double sin(double x): 求 x (弧度表示)的正弦值。
- double tan(double x): 求 x (弧度表示)的正切值。

③其他

- double hypot(double x, double y): x、y 为直角三角形的两个直角边, 返回其斜边的长度。
- double modf(double x, double *ip): 将 x 的整数部分通过指针回传, 返回其小数部分, 并将整数部分保存在 *ip 中。

5 ctype.h

如果参数 c 满足函数条件, 则如下函数返回非零值(即 true)。如果不满足, 则如下函数返回零(即 false)。

- int isalnum(int c): 检查所传的字符是否是字母和数字。
- int isalpha(int c): 该函数检查所传的字符是否是字母。
- int islower(int c): 检查所传的字符是否是小写字母。
- int isupper(int c): 检查所传的字符是否是大写字母。
- int isdigit(int c): 检查所传的字符是否是十进制数字。
- int isxdigit(int c): 检查所传的字符是否是十六进制数字。
- **int tolower(int c): 把大写字母转换为小写字母。**
- **int toupper(int c): 把小写字母转换为大写字母。**

附录

- 我的个人博客: messi1002.top
- 如有错误或疑惑之处 请联系 wjymessi@163.com

