1.升序与降序排列

```c
#include <bits/stdc++.h>
int compare_asc(const void *a,const void *b){
    return (*(int*)a-*(int*)b);
}
int main(){
    int arr[] = {5,2,9,1,5,6};
    int n = sizeof(arr)/sizeof(arr[0]);
    qsort(arr,n,sizeof(int),compare_asc);
    for(int i=0;i<n;i++){
        printf("%d",arr[i]);
    }
}
```

2.字符串CRUD

```c
#include <bits/stdc++.h>  // 万能头文件，包含了C标准库中的大部分头文件

int main() {
    // 初始化字符串
    char str[100] = "Hello, World!";
    printf("原始字符串: %s\n", str);

    // 1. 字符串长度 - strlen
    // 功能：计算字符串的长度，不包括终止符 '\0'
    size_t len = strlen(str);
    printf("字符串长度: %zu\n", len);

    // 2. 字符串复制 - strcpy
    // 功能：将源字符串复制到目标字符串中，包括终止符 '\0'
    char dest[100];
    strcpy(dest, str);
```

```c
    printf("复制后的字符串: %s\n", dest);

    // 3. 字符串连接 - strcat
    // 功能：将源字符串连接到目标字符串的末尾，覆盖目标字符串的终止符 '\0'
    strcat(dest, " Welcome to C!");
    printf("连接后的字符串: %s\n", dest);

    // 4. 字符串比较 - strcmp
    // 功能：比较两个字符串，返回值为0表示相等，小于0表示str1小于str2，大于0表示str1大于str2
    int cmp = strcmp(str, dest);
    if (cmp == 0) {
        printf("两个字符串相等\n");
    } else if (cmp < 0) {
        printf("str 小于 dest\n");
    } else {
        printf("str 大于 dest\n");
    }

    // 5. 字符串查找 - strchr
    // 功能：在字符串中查找指定字符的第一次出现的位置，返回指向该位置的指针，未找到返回NULL
    char *ptr = strchr(str, 'W');
    if (ptr != NULL) {
        printf("找到字符 'W'，位置: %ld\n", ptr - str);
    } else {
        printf("未找到字符 'W'\n");
    }

    // 6. 字符串查找 - strstr
    // 功能：在字符串中查找指定子串的第一次出现的位置，返回指向该位置的指针，未找到返回NULL
    ptr = strstr(str, "World");
    if (ptr != NULL) {
        printf("找到子串 'World'，位置: %ld\n", ptr - str);
    } else {
        printf("未找到子串 'World'\n");
    }

    // 7. 字符串分割 - strtok
    // 功能：将字符串分割为一系列子串，第一次调用时传入字符串，后续调用传入NULL
    char str2[] = "This,is,a,test";
```

```
    char *token = strtok(str2, ",");
    while (token != NULL) {
        printf("分割后的子串: %s\n", token);
        token = strtok(NULL, ",");
    }

    // 8. 字符串填充 - memset
    // 功能：将指定内存区域的前n个字节设置为特定值
    memset(str, 'A', 5);
    printf("填充后的字符串: %s\n", str);

    // 9. 字符串复制指定长度 - strncpy
    // 功能：将源字符串的前n个字符复制到目标字符串中，如果源字符串长度小于n，则用 '\0' 填充
    char dest2[100];
    strncpy(dest2, str, 5);
    dest2[5] = '\0';  // 手动添加终止符
    printf("复制指定长度后的字符串: %s\n", dest2);

    // 10. 字符串比较指定长度 - strncmp
    // 功能：比较两个字符串的前n个字符，返回值为0表示相等，小于0表示str1小于str2，大于0表示str1大于str2
    cmp = strncmp(str, dest2, 5);
    if (cmp == 0) {
        printf("前5个字符相等\n");
    } else if (cmp < 0) {
        printf("str 的前5个字符小于 dest2\n");
    } else {
        printf("str 的前5个字符大于 dest2\n");
    }

    // 11. 字符串查找字符最后一次出现的位置 - strrchr
    // 功能：在字符串中查找指定字符的最后一次出现的位置，返回指向该位置的指针，未找到返回NULL
    ptr = strrchr(str, 'A');
    if (ptr != NULL) {
        printf("找到字符 'A' 的最后一次出现，位置: %ld\n", ptr - str);
    } else {
        printf("未找到字符 'A'\n");
    }

    // 12. 字符串查找字符集中任意字符的第一次出现 - strpbrk
```

```
    // 功能：在字符串中查找字符集中任意字符的第一次出现的位置，返回指向该位置的指针，未找到返回NULL
    ptr = strpbrk(str, "AB");
    if (ptr != NULL) {
        printf("找到字符集 'AB' 中的字符，位置: %ld\n", ptr - str);
    } else {
        printf("未找到字符集 'AB' 中的字符\n");
    }

    // 13. 字符串查找字符集中任意字符的第一次出现的位置 - strspn
    // 功能：返回字符串开头连续包含字符集中字符的字符数
    size_t span = strspn(str, "A");
    printf("字符串开头连续包含字符 'A' 的字符数: %zu\n", span);

    // 14. 字符串查找字符集中任意字符的第一次不匹配的位置 - strcspn
    // 功能：返回字符串开头连续不包含字符集中字符的字符数
    span = strcspn(str, "B");
    printf("字符串开头连续不包含字符 'B' 的字符数: %zu\n", span);

    return 0;
}
```

3.类型判别

```
#include <bits/stdc++.h>  // 万能头文件，包含了C标准库中的大部分头文件

int main() {
    char ch;

    // 1. isalnum - 检查字符是否为字母或数字
    // 功能：如果字符是字母（A-Z, a-z）或数字（0-9），返回非零值；否则返回0
    ch = 'A';
    if (isalnum(ch)) {
        printf("'%c' 是字母或数字\n", ch);
    } else {
        printf("'%c' 不是字母或数字\n", ch);
    }
```

```
    // 2. isalpha - 检查字符是否为字母
    // 功能：如果字符是字母（A-Z, a-z），返回非零值；否则返回0
    ch = 'z';
    if (isalpha(ch)) {
        printf("'%c' 是字母\n", ch);
    } else {
        printf("'%c' 不是字母\n", ch);
    }

    // 3. isdigit - 检查字符是否为数字
    // 功能：如果字符是数字（0-9），返回非零值；否则返回0
    ch = '9';
    if (isdigit(ch)) {
        printf("'%c' 是数字\n", ch);
    } else {
        printf("'%c' 不是数字\n", ch);
    }

    // 4. isxdigit - 检查字符是否为十六进制数字
    // 功能：如果字符是十六进制数字（0-9, A-F, a-f），返回非零值；否则返回0
    ch = 'F';
    if (isxdigit(ch)) {
        printf("'%c' 是十六进制数字\n", ch);
    } else {
        printf("'%c' 不是十六进制数字\n", ch);
    }

    // 5. islower - 检查字符是否为小写字母
    // 功能：如果字符是小写字母（a-z），返回非零值；否则返回0
    ch = 'm';
    if (islower(ch)) {
        printf("'%c' 是小写字母\n", ch);
    } else {
        printf("'%c' 不是小写字母\n", ch);
    }

    // 6. isupper - 检查字符是否为大写字母
    // 功能：如果字符是大写字母（A-Z），返回非零值；否则返回0
    ch = 'Z';
    if (isupper(ch)) {
        printf("'%c' 是大写字母\n", ch);
    } else {
```

```
        printf("'%c' 不是大写字母\n", ch);
    }

    // 7. isspace - 检查字符是否为空白字符
    // 功能：如果字符是空白字符（空格、制表符、换行符等），返回非零值；否则返回0
    ch = '\t';
    if (isspace(ch)) {
        printf("'%c' 是空白字符\n", ch);
    } else {
        printf("'%c' 不是空白字符\n", ch);
    }

    // 8. iscntrl - 检查字符是否为控制字符
    // 功能：如果字符是控制字符（ASCII 0-31 和 127），返回非零值；否则返回0
    ch = '\n';
    if (iscntrl(ch)) {
        printf("'%c' 是控制字符\n", ch);
    } else {
        printf("'%c' 不是控制字符\n", ch);
    }

    // 9. ispunct - 检查字符是否为标点符号
    // 功能：如果字符是标点符号（非字母、数字、空白字符或控制字符），返回非零值；否则返回0
    ch = '!';
    if (ispunct(ch)) {
        printf("'%c' 是标点符号\n", ch);
    } else {
        printf("'%c' 不是标点符号\n", ch);
    }

    // 10. isprint - 检查字符是否为可打印字符
    // 功能：如果字符是可打印字符（包括空格，但不包括控制字符），返回非零值；否则返回0
    ch = '$';
    if (isprint(ch)) {
        printf("'%c' 是可打印字符\n", ch);
    } else {
        printf("'%c' 不是可打印字符\n", ch);
    }

    // 11. isgraph - 检查字符是否为图形字符
    // 功能：如果字符是图形字符（可打印字符，不包括空格），返回非零值；否则返回0
```

```
    ch = '#';
    if (isgraph(ch)) {
        printf("'%c' 是图形字符\n", ch);
    } else {
        printf("'%c' 不是图形字符\n", ch);
    }

    // 12. tolower - 将字符转换为小写
    // 功能：如果字符是大写字母（A-Z），返回对应的小写字母；否则返回原字符
    ch = 'G';
    char lower_ch = tolower(ch);
    printf("'%c' 转换为小写后是 '%c'\n", ch, lower_ch);

    // 13. toupper - 将字符转换为大写
    // 功能：如果字符是小写字母（a-z），返回对应的大写字母；否则返回原字符
    ch = 'h';
    char upper_ch = toupper(ch);
    printf("'%c' 转换为大写后是 '%c'\n", ch, upper_ch);

    return 0;
}
```

4.数学函数

```
#include <stdio.h>
#include <math.h>  // 数学函数库

int main() {
    // 浮点数操作
    double x = 2.5;
    double y = 3.7;

    // 1. 平方根 - sqrt
    // 功能：计算一个数的平方根
    double sqrt_val = sqrt(x);
    printf("sqrt(%.2f) = %.2f\n", x, sqrt_val);

    // 2. 幂运算 - pow
```

```c
    // 功能：计算 x 的 y 次方
    double pow_val = pow(x, y);
    printf("pow(%.2f, %.2f) = %.2f\n", x, y, pow_val);

    // 3. 指数函数 - exp
    // 功能：计算 e 的 x 次方
    double exp_val = exp(x);
    printf("exp(%.2f) = %.2f\n", x, exp_val);

    // 4. 自然对数 - log
    // 功能：计算 x 的自然对数（以 e 为底）
    double log_val = log(x);
    printf("log(%.2f) = %.2f\n", x, log_val);

    // 5. 常用对数 - log10
    // 功能：计算 x 的常用对数（以 10 为底）
    double log10_val = log10(x);
    printf("log10(%.2f) = %.2f\n", x, log10_val);

    // 6. 绝对值 - fabs
    // 功能：计算浮点数的绝对值
    double fabs_val = fabs(-x);
    printf("fabs(-%.2f) = %.2f\n", x, fabs_val);

    // 7. 向上取整 - ceil
    // 功能：返回大于或等于 x 的最小整数
    double ceil_val = ceil(x);
    printf("ceil(%.2f) = %.2f\n", x, ceil_val);

    // 8. 向下取整 - floor
    // 功能：返回小于或等于 x 的最大整数
    double floor_val = floor(x);
    printf("floor(%.2f) = %.2f\n", x, floor_val);

    // 9. 四舍五入 - round
    // 功能：返回最接近 x 的整数
    double round_val = round(x);
    printf("round(%.2f) = %.2f\n", x, round_val);

    // 10. 三角函数 - sin, cos, tan
    // 功能：计算 x 的正弦、余弦和正切值（x 为弧度）
    double sin_val = sin(x);
```

```c
    double cos_val = cos(x);
    double tan_val = tan(x);
    printf("sin(%.2f) = %.2f\n", x, sin_val);
    printf("cos(%.2f) = %.2f\n", x, cos_val);
    printf("tan(%.2f) = %.2f\n", x, tan_val);

    // 11. 反三角函数 - asin, acos, atan
    // 功能：计算 x 的反正弦、反余弦和反正切值（结果为弧度）
    double asin_val = asin(0.5);
    double acos_val = acos(0.5);
    double atan_val = atan(1.0);
    printf("asin(0.5) = %.2f\n", asin_val);
    printf("acos(0.5) = %.2f\n", acos_val);
    printf("atan(1.0) = %.2f\n", atan_val);

    // 12. 双曲函数 - sinh, cosh, tanh
    // 功能：计算 x 的双曲正弦、双曲余弦和双曲正切值
    double sinh_val = sinh(x);
    double cosh_val = cosh(x);
    double tanh_val = tanh(x);
    printf("sinh(%.2f) = %.2f\n", x, sinh_val);
    printf("cosh(%.2f) = %.2f\n", x, cosh_val);
    printf("tanh(%.2f) = %.2f\n", x, tanh_val);

    // 13. 浮点数取余 - fmod
    // 功能：计算 x 除以 y 的余数
    double fmod_val = fmod(x, y);
    printf("fmod(%.2f, %.2f) = %.2f\n", x, y, fmod_val);

    // 定点数操作（使用整数模拟）
    int a = 5;
    int b = 2;

    // 14. 绝对值 - abs
    // 功能：计算整数的绝对值
    int abs_val = abs(-a);
    printf("abs(-%d) = %d\n", a, abs_val);

    // 15. 取余 - %
    // 功能：计算 a 除以 b 的余数
    int mod_val = a % b;
    printf("%d %% %d = %d\n", a, b, mod_val);
```

```c
    // 16. 最大值和最小值 - fmax, fmin
    // 功能：返回两个浮点数中的最大值或最小值
    double max_val = fmax(x, y);
    double min_val = fmin(x, y);
    printf("fmax(%.2f, %.2f) = %.2f\n", x, y, max_val);
    printf("fmin(%.2f, %.2f) = %.2f\n", x, y, min_val);

    // 17. 浮点数比较 - isgreater, isless
    // 功能：比较两个浮点数的大小，返回布尔值
    if (isgreater(x, y)) {
        printf("%.2f 大于 %.2f\n", x, y);
    } else {
        printf("%.2f 不大于 %.2f\n", x, y);
    }

    if (isless(x, y)) {
        printf("%.2f 小于 %.2f\n", x, y);
    } else {
        printf("%.2f 不小于 %.2f\n", x, y);
    }

    return 0;
}
```