1. 基本数据类型、表达式、输入输出

1.1掌握基本数据类型的定义与操作；熟练编程语言的输入输出方法；

1.2 能够正确运用表达式，熟悉运用算术表达式、关系表达式、逻辑表达式；

1.3 了解输入输出的格式控制。

2.算法与控制结构

掌握各种运算符的使用；掌握各种过程化控制语句结构；

2.1 理解算法的概念

2.2 熟练运用条件结构进行编程；

2.2 熟练运用循环结构进行编程。

2.3 能够根据问题需要设计解决的算法

3. 函数

3.1掌握函数声明与定义；

3.2掌握全局变量、静态变量、局部变量；

3.3 掌握递归函数等的定义与调用方法。

4. 数组

掌握数组的定义与使用；掌握数组做函数参数的用法；掌握数组各种排序和**查找**算法。

冒泡

int a[5] = { 5,2,4,3,1 };

for (int i = 0; i < 4; i++) {

bool flag = false;

for (int j = 4; j > i; j--) {

if (a[j] < a[j - 1]) {

exchange(a[j], a[j - 1]);

flag = true;

}

}

if (flag == false)

break;

}

快速排序

int Partition(int arr[], int low, int high) {

int pivot = arr[low];

while (low<high)

{

while (low < high && arr[high] >= pivot)

high--;

arr[low] = arr[high];

while (low < high && arr[low] <= pivot)

low++;

arr[high] = arr[low];

}

arr[low] = pivot;

return low;

}

void QuickSort(int arr[], int low, int high) {

if (low < high) {

int pivotPos = Partition(arr, low, high);

QuickSort(arr, low, pivotPos - 1);

QuickSort(arr, pivotPos + 1, high);

}

}

查找

int BinarySearch(int arr[], int low, int high, int key) {

while (low <= high) {

int mid = (low + high) / 2;

if (arr[mid] == key)

return mid;

if (arr[mid] < key)

low = mid + 1;

if (arr[mid] > key)

high = mid - 1;

}

return -1;

}

4.1 数组定义和初始化数组；

4.2 数组做函数参数；

4.3 二维数组；

初始化：

//方法一

int arr1[2][3];

//方法二

int arr2[2][3] = { {1,2,3},{4,5,6} };

//方法三

int arr3[2][3] = { 1,2,3,4,5,6 };

//方法四

int arr4[][3] = { 1,2,3,4,5,6 };

使用操作：

int arr[2][3] = { {1,2,3},{4,5,6} };

cout << "二维数组所占内存大小：" << sizeof(arr) << endl;

cout << "二维数组数组元素个数：" << sizeof(arr) / sizeof(arr[0][0]) << endl;

cout << "数组首地址：" << (int)arr << endl;

cout << "数组第一个元素地址：" << (int)&arr[0][0] << endl;

cout << "数组第一行地址：" << (int)(arr[0]) << endl;

cout << "数组第二行地址：" << (int)(arr + 1) << endl;

cout << "数组第二行地址：" << (int)(arr[1]) << endl;

cout << "数组第二个元素地址（a[0][1]）：" << (int)&arr[0][1] << endl;

//首地址+4\*列\*行

cout << "下一地址（注意与数组首地址区分）：" << (int)(&arr + 1) << endl;

4.4 字符数组与字符串，字符串函数。

5. 结构体与类

5.1了解结构体及其成员的定义与使用；

5.2 掌握类的定义及成员函数的定义和使用；

6. 文件读写

掌握文件读写的方法，根据问题需要从文件读取数据和保存数据到文件。

6.1 掌握文件的读写模式；

6.2掌握文本文件的读写；

6.3 掌握二进制文件的读写；

参考：<https://blog.csdn.net/weixin_44190648/article/details/122358591>

7. 异常处理

掌握编程语言异常处理的机制，会在编程中运用异常处理。

参考：<https://blog.csdn.net/weixin_44190648/article/details/122322641>

1.默认参数

2.占位参数：重载i++、++i

3.深拷贝、浅拷贝

4.静态成员变量、静态成员函数

静态变量再静态区分配不空间，不在堆栈分配空间。只有等到程序全部运行结束才会释放空间

5.friend友元函数、友元类

6.运算符重载

7.const

8.多继承、虚继承

