# 点和箭头运算符

点运算符( . )的左边操作数是一个结果为结构的表达式；

箭头运算符( -> )的左边的操作数是一个指向结构体的指针。

一般对象用.，指针对象用->。

typedef struct // 定义一个结构体类型：DATA

{

char key[10]; // 结构体成员：key

char name[20]; // 结构体成员：name

int age; // 结构体成员：age

}DATA;

DATA data; // 声明一个结构体变量

DATA \*pdata; // 声明一个指向结构体的指针

// 访问数据操作如下：

data.age = 24; // 结构体变量通过点运算符( . )访问

pdata->age = 24; // 指向结构体的指针通过箭头运算符( -> )访问

# c++ sort函数

sort函数可以三个参数也可以两个参数，必须的头文件#include < algorithm>和using namespace std;

Sort函数有三个参数：（第三个参数可不写）

（1）第一个是要排序的数组的起始地址。

（2）第二个是结束的地址（最后一位要排序的地址）

（3）第三个参数是排序的方法，可以是从大到小也可是从小到大，还可以不写第三个参数，此时默认的排序方法是从小到大排序。

#include <iostream>

#include <algorithm>

int main()

{

int a[20]={2,4,1,23,5,76,0,43,24,65},i;

sort(a,a+20);

for(i=0;i<20;i++)

cout<<a[i]<<endl;

return 0;

}

Sort函数的第三个参数可以用这样的语句告诉程序你所采用的排序原则

less<数据类型>()//从小到大排序

greater<数据类型>()//从大到小排序

sort(a,a+10,less<int>());

sort(a,a+10,greater<char>());

# new和delete

**变量**

int \*a = new int;  
delete a;   //释放单个int的空间

new 结构体名/类型()；

**一维数组**

int \*a = new int[5];  
初始化：memset(array,0,n\*sizeof(array)); (也可以利用一个for循环对其赋值初始化)

Memset按字节填充，int型填0为0，填1不为1，char型填0为\0

delete []a;    //释放int数组空间

**二维数组**

int \*\*array2D;

//假定数组第一维长度为m， 第二维长度为n

//动态分配空间

array2D = new int \*[m];

for( int i=0; i<m; i++ )

{

array2D[i] = new int [n] ;

}

//释放

for( int i=0; i<m; i++ )

{

delete [] array2D[i];

}

delete [] array2D;

int n;

cin >> n;

int \*array;

array = new int [n];

memset(array, 0, n \* sizeof(int));

new时n可以为变量，直接声明数组不可以