#include<iostream>

using namespace std;

//实验二众数问题

//分治算法解决众数求解（数组已经按从小到大排好序，如果不止一个众数则只显示其中一个众数的值）

void solve(int a[],int n,int &l,int &r){

int mid=n/2;

l=mid-1;

r=mid+1;//记录a[mid]的起始位置和终止位置

while(l>=0&&a[mid]==a[l]){

l--;

}

l++;

while(r<n&&a[r]==a[mid]){

r++;

}

r--;

}

void Solution(int a[],int n,int &mid,int &maxcnt){

if(n==1){

mid=0;

maxcnt=1;

}

int l=0,r=0;

solve(a,n,l,r);

int num=n/2;

int cnt=r-l+1;

if(cnt>maxcnt){

maxcnt=cnt;

mid=a[num];

}

if(l>maxcnt){

Solution(a,l--,mid,maxcnt);

}

if(n-r>maxcnt){

Solution(a+r,n-r,mid,maxcnt);

}

}

int main(){

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

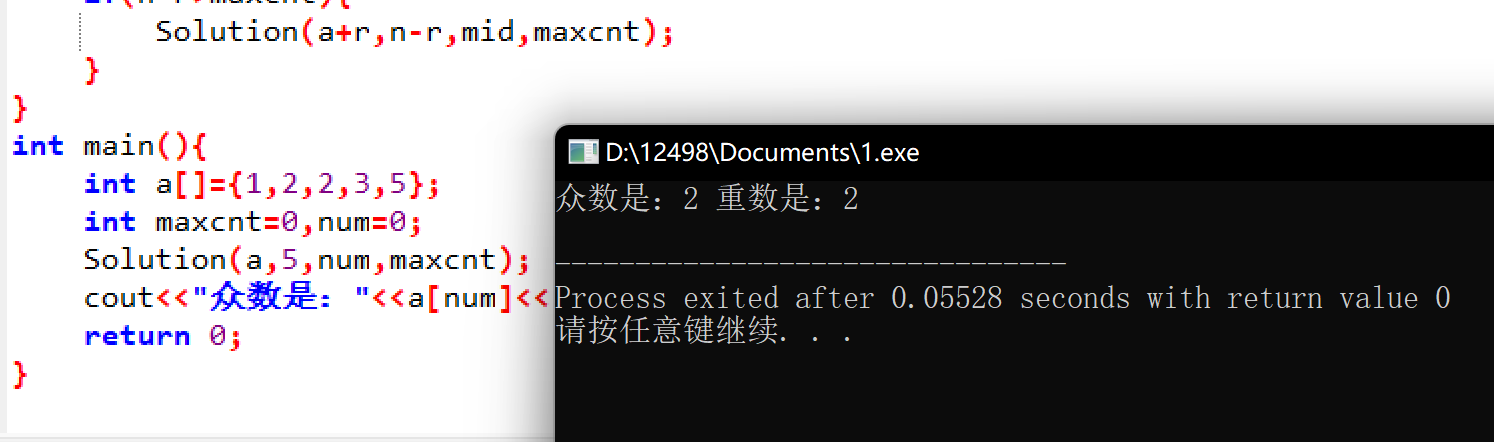
int maxcnt=0,num=0;

Solution(a,5,num,maxcnt);

cout<<"众数是："<<a[num]<<" 重数是："<<maxcnt<<endl;

return 0;

}



//实验五

//求解一个整数数组划分为两个子数组的问题：分成两个数组个数相差最少，和相差最大

//利用快速排序思想

#include<iostream>

using namespace std;

int Partition(int a[],int l,int r){

int i=l,j=r;

int temp=a[l];//以最左边元素为基准点

while(i!=j){

while(j>i&&a[j]>=temp)j--;

a[i]=a[j];

while(j>i&&a[i]<=temp)i++;

a[j]=a[i];

}

a[i]=temp;

return i;

}

int solve(int a[],int n){

int l=0,r=n-1;

int i=Partition(a,l,r);

while(1){

if(i==n/2)break;

else if(i>n/2){

i=Partition(a,0,i-1);

}

else{

i=Partition(a,i+1,r);

}

}

int s1=0,s2=0;

for(int i=0;i<n/2;i++)//算左半部分的和

s1+=a[i];

for(int j=n/2;j<n;j++)//算右半部分的和

s2+=a[j];

return s2-s1;

}

int main(){

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

cout<<"结果为："<<solve(a,5);

return 0;

}

