测试代码

#include <stdio.h>

#include <stdlib.h>

#include<time.h>

#define SIZE 16

void setRandomArray(int arr[] );

void sortArraySelect(int arr[]);

void displaySumSelect(int arr[]);

int main()

{

int i=0;

int Arr[SIZE]={0};

setRandomArray(Arr);//生成随机数组

sortArraySelect(Arr);//将其升序排列

for(i=0;i<SIZE;i++)//打印排列好的数组

{

printf("%4d",Arr[i]);

}

printf("\n");

displaySumSelect(Arr);//获得该数组中有和关系的集合并打印

return 0;

}

void setRandomArray(int arr[])//生成随机数组

{

srand(time(NULL));

int i=0;

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

{

arr[i]=rand()%99+1;

}

}

void sortArraySelect(int arr[])//利用选择排序使数组升序排列

{

int i=0;

int j=0;

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

{

for(j=i+1;j<SIZE;j++)

{

int temp=arr[j];

if(arr[j]<arr[i])

{

arr[j]=arr[i];

arr[i]=temp;

}

}

}

}

void displaySumSelect(int arr[])//打印存在两数之和等于第三个数关系的集合

{

int i=0;int j=1;int k=2;

for(i=0;i<SIZE-2;i++)

{

for(j=i+1;j<SIZE-1;j++)

{

for(k=j+1;k<SIZE;k++)

{

if(arr[i]+arr[j]==arr[k])

{

printf("{%d,%d,%d}\n",arr[i],arr[j],arr[k]);

}

}

}

}

}

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方法 | 测试结果 | 测试结论 |
| 1 | 测试setRandomArray函数能否正常生成随机数组 | 输入正常值和异常值，考察程序是否报错 |  | 三次分别生成了不同的随机数组如图所示，说明该函数可以正常运行 | 通过 |  |
| 2 | 测试sortArraySelect函数是否可以正常使用使数组升序排列 | 输入正常值和异常值，考察程序是否报错 |  | 三次生成不同的随机数组在经函数操作后可以正常将数组升序排列，说明函数可以正常运行 | 通过 |  |
| 3 | 测试displaySumRelation函数是否可以正常使用 | 输入正常值和异常值，考察程序是否报错 |  | 三次生成不同的随机数组在经过排列后成功找到了有相加关系的集合，说明程序可以正常运行 | 通过 |  |
|  |  |  |  |  |  |  |

测试结论：

该题目所有要求均完成