测试报告

|  |  |  |
| --- | --- | --- |
| 崔曼妮 | U202413702 | B01 |

测试代码：

//崔曼妮

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define SIZE 20

void setRandomArray(int\* random,const int size);

void bubbleSort(int\* x,const int size);

void displayResult(int\* x,const int size,int \*foundNum);

int main()

{

srand((unsigned int)time(NULL));

int random[SIZE];

setRandomArray(random,SIZE);

bubbleSort(random,SIZE);

int foundNum = 0;

displayResult(random,SIZE,&foundNum);

printf("\nthe number of array is %d\n",foundNum);

return 0;

}

void setRandomArray(int\* random,const int size)

{

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

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

}

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

printf("%d\t",random[i]);

}

}

void bubbleSort(int\* x,const int size)

{

int i,j,isSwapped,hold;

for (i = 1;i < size;i++){

isSwapped = 0;

for (j = 0;j < size - i;j++){

if(x[j] > x[j+1]){

hold = x[j];

x[j] = x[j + 1];

x[j + 1] = hold;

isSwapped++;

}

}

if(isSwapped == 0){

break;

}

}

printf("\nafter sort:");

for (int k = 0;k < size;k++){

printf("%d\t",x[k]);

}

}

void displayResult(int\* x,const int size,int \*foundNum)

{

//从第一个开始找

for (int i = 0;i < size - 2;i++){

int count = 0;

int the\_minus = x[i+1]-x[i];

for (int j = i + 1;j < size;j++){

if ((x[j + 1]-x[j]) == the\_minus){

count++;

}

else{

break;

}

}

//用while循环确保如果长度大于三的话以第i个开头的数列既能打印出长度为4个及以上的，也能打出其包含的长度是3个的数列

while(count >= 1){

(\*foundNum)++;

printf("\n{%d,",x[i]);

for (int j = 0;j < count;j++){

printf("%d,",x[i + j + 1]);

}

printf("%d}",x[i + count + 1]);

count--;

}

}

}

测试过程：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方法 | 测试结果 | 测试结论 |
| 1 | void setRandomArray(int\* random,const int size)生成随机整数 | 生成并打印随机整数 |  | 完成 |
| 2 | void setRandomArray(int\* random,const int size)  冒泡排序 | 打印排序后的数 |  | 完成 |
| 3 | void displayResult(int\* x,const int size,int \*foundNum)打印所有等差数列 | 多次生成随机数测试 |  | 完成 |

测试结论：

三个任务均完成