# 机考测试报告

姓名： 刘欣 学号： U202410809 题号： B03

【测试代码】

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define SIZE 16 // 定义数组长度

// 设置数组为随机整数 , 元素取值范围为1-99

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

// 对数组升序排序，选择排序

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

// 查找是否存在两数和的关系 ， 找到后打印 ，指针返回找到的个数

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

// 后面删掉

void disPlay( int array[ ] , int size ) ;

int main()

{

// 设置数组为随机整数 , 元素取值范围为1-99

printf("Set random array : \n") ;

srand(time(NULL)) ; // 随机种子

int array[SIZE] ; // 定义数组

setRandomArray(array , SIZE) ; // 产生随机数组

disPlay(array , SIZE ) ; // 展示初始未排序数组

printf("\n") ;

// 对数组升序排序，选择排序

printf("Display the sorted array : \n") ;

sortArraySelect(array , SIZE) ;

disPlay(array , SIZE ) ; // 展示排序数组

printf("\n") ;

// 查找是否存在两数和的关系 ， 找到后打印 ，指针返回找到的个数

printf("Display the sum relation :\n");

int foundNum = 0 ;

displaySumRelation(array , SIZE , &foundNum ) ;

printf("\n%d relations in sum\n",foundNum) ;

return 0;

}

// 设置数组为随机整数 , 元素取值范围为1-99

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

{

int i ;

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

{

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

}

}

// 对数组升序排序，选择排序

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

{

int i , j , min\_index , min\_value ;

for ( i = 0 ; i < size - 1 ; i ++ )

{

min\_index = i ;

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

{

if ( x[j] < x[min\_index])

{

min\_index = j ;

}

}

if (min\_index != i)

{

min\_value = x[min\_index] ;

x[min\_index] = x[i] ;

x[i] = min\_value ;

}

}

}

// 查找是否存在两数和的关系 ， 找到后打印 ，指针返回找到的个数

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

{

int i , j , k ;

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

{

for ( j = 0 ; j < size ; j ++ )

{

if ( j < i )

{

break ; // 减少计算机计算，提高效率

}

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

{

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

{

printf("%d + %d = %d \n",x[i] , x[j] , x[k] ) ;

(\*foundNum) ++ ;

}

}

}

}

}

void disPlay( int array[ ] , int size )

{

int i ;

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

{

printf("%2d " , array[i]);

if ( ( i + 1) % 4 == 0 )

{

printf("\n") ;

}

}

}

【测试过程】

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方法 | 测试结果 | 测试结论 |
| 1 | 测试setRandomArray函数，考察程序是否报错 | 多次调用该函数，考察程序是否报错 |  | 测试通过 |
| 2 | 测试sortArraySelect函数，考察程序是否报错 | 多次调用该函数，考察程序是否报错 |  | 测试通过 |
| 3 | 测试displaySumRelation函数，考察程序是否报错 | 多次调用该函数，考察程序是否报错 |  | 测试通过 |

【测试结论】

该题所有要求都完成。