机考测试报告

姓名：陈泽州 学号：U202411198 题号：C02

**【测试代码】**

/\*

Authored by Zezhou Chen

U202411198

\*/

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define M 10

#define N 35

typedef struct

{

char content[N];

int length;

} String;

void sort\_insert( String \*array , const int arraySize );

//使用插入排序进行排序

int check( const String \*x , const String \*y );

//检查x是否应该放y前面 如果是返回1 否则返回0

void swap( String \*x , String \*y );

//交换x和y

void strcpy( const String \*origin , String \*target);

//将origin拷贝到target上面

void displayContent( String \*x );

//打印x的content

void setRandomString( String \*x );

//随机生成一串长度随机的数列

void printArray(String \*array , const int arraySize );

//打印所有数列

int main ( void )

{

srand(time(NULL));

String \*array = malloc( sizeof(String) \* M );

//申请一个空间

int i;

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

{

setRandomString(array+i);

}

printf("generte %d random array :\n" , M);

printArray(array , M);

sort\_insert(array , M );

printf("After sort:\n");

printArray(array , M);

free(array);

return 0;

}

void sort\_insert( String \*array , const int arraySize )

{

int i , j;

for( i = 1 ; i < arraySize ; i ++ )

{

for( j = i ; j > 0 ; j-- )

{

if( check(array+j-1 , array+j) )

{

break;

}

else

{

swap(array+j , array+j-1);

}

}

}

}

void displayContent( String \*x )

{

int i;

for( i = 0 ; i < x->length ; i ++ )

{

printf("%c" , x->content[i] );

}

printf("\n");

}

int check( const String \*x , const String \*y )

{

if( x->content[0] != y->content[0] )

{

return x->content[0] <= y->content[0];

}

return x->length <= y->length;

}

void strcpy( const String \*origin , String \*target)

{

target->length = origin->length;

int i;

for( i = 0 ; i < target->length ; i ++ )

{

target->content[i] = origin->content[i];

}

}

void swap( String \*x , String \*y )

{

String \*temp = malloc(sizeof(String));

strcpy( x , temp );

strcpy( y , x );

strcpy ( temp , y );

}

void setRandomString( String \*x )

{

x->length = rand() % N;

int i;

for( i = 0 ; i < x->length ; i ++ )

{

x->content[i] = rand() % 10 + '0';

}

if(x->length == 0)

{

x->content[0] = 'a';

}

}

void printArray(String \*array , const int arraySize )

{

int i;

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

{

printf("[%d]:" , i);

displayContent(array+i);

}

}

**【测试过程】**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方法 | 测试结果 | 测试结论 |
| 1 | 测试能否随机生成M个字符串并排序 | 随机生成3组，观察结果 |  | 测试通过 |
| 2 | 测试setRandomString函数，考察其能否产生随机长度的小写字母串 | 随机生成3组观察结果 |  | 测试通过 |
| 3 | 测试sort\_insert能否按要求进行排序 | 随机生成3组进行观察测试 |  | 测试通过 |

**【测试结论】**

该题所有要求都完成