# 机考测试报告

姓名： 刘欣 学号： U202410809 题号： C04

【测试代码】

// 刘欣 ，电创2401班

// U202410809

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#define N 35 // 字符串的最大长度

typedef struct

{

char content[N] ; // 字符串内容

int length ; // 字符串的有效长度

}String ;

// 设置字符串内容 ， 长度为0—N的随机值

void setRandomString(String \*s);

// 得到所求交集对象

void getStringIntersection(String \*s1 , String \*s2 , String \*result ) ;

// 展示字符串的字符和长度

void displayString( String s ) ;

int main()

{

// 创建两个字符串结构体对象，随机设置内容

srand(time(NULL)) ;

String string1 ; // 初始化字符串结构体

String string2 ;

setRandomString(&string1) ; // 设置字符串内容 ， 长度为0—N(35)的随机值

setRandomString(&string2) ;

displayString(string1) ; // 展示字符串数组

displayString(string2) ;

// 调用函数得到所求交集对象

String result ;

getStringIntersection(&string1 , &string2 , &result) ;

displayString(result) ; // 展示交集

return 0;

}

// 设置字符串内容 ， 长度为0—N的随机值

void setRandomString(String \*s)

{

s->length = rand( ) % N ;

int i ;

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

{

if (rand( ) % 2 == 0) // 设定为小写

{

s->content[i] = 'a' + rand( ) % 26 ;

}

else // 设定为大写

{

s->content[i] = 'A' + rand( ) % 26 ;

}

}

}

// 得到所求交集对象

void getStringIntersection(String \*s1 , String \*s2 , String \*result )

{

int a[256] = {0} ; // 作为标记看两数组的相同字母，ASCII码表不会超过256

int i ;

int k = 0 ;

result->length = 0 ;

for ( i = 0 ; i < s1->length ; i ++ ) // 先标记第一遍

{

a[s1->content[i]] = 1 ; // 利用ASCII表的数字

}

for ( i = 0 ; i < s2->length ; i ++ ) // 再查看是否有相同标记

{

if ( a[s2->content[i]] == 1 ) // 说明两数组有相同元素

{

result->content[k++] = s2->content[i] ;

result->length ++ ;

}

}

}

// 展示字符串的字符和长度

void displayString( String s )

{

int i ;

for ( i = 0 ; i < s.length ; i ++ )

{

printf("%c",s.content[i]) ;

}

printf("\nThe string has total %d alp\n\n",s.length) ;

}

【测试过程】

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 测试任务 | 测试方法 | 测试结果 | 测试结论 |
| 1 | main函数创建两个字符串结构体变量，随机设置其内容，并求其交集并打印。 | 多次run and build ,考察程序是否报错 |  | 测试通过 |
| 2 | 测试setRandomString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getStringIntersect  ion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。