

清华大学本科生考试试题专用纸

考试课程 计算机语言与程序设计 (卷) 2013 年 12 月 10 日

姓名_____ 班级_____ 学号_____

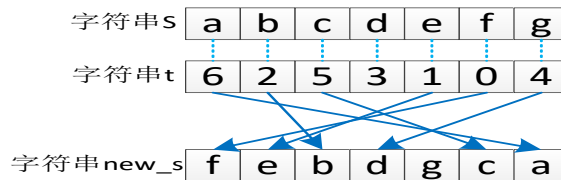
大题	题号	一	二	三	总分
	分数	30 分	35 分	35 分	
成绩					

试卷提交说明

- 在考卷上填写你的姓名、学号；
- 按要求编程，调试运行，将每道题的结果考屏，粘贴在考卷对应的题后；
- 交卷时，提交一个压缩文件包，包括：
 - 文件包名
学号+姓名，如：2012123456 张三.rar
 - 文件包内容
 - 每道题的 c 工程文件夹；
 - 考卷。

一、字符串映射。编制一个程序，要求实现如下功能：

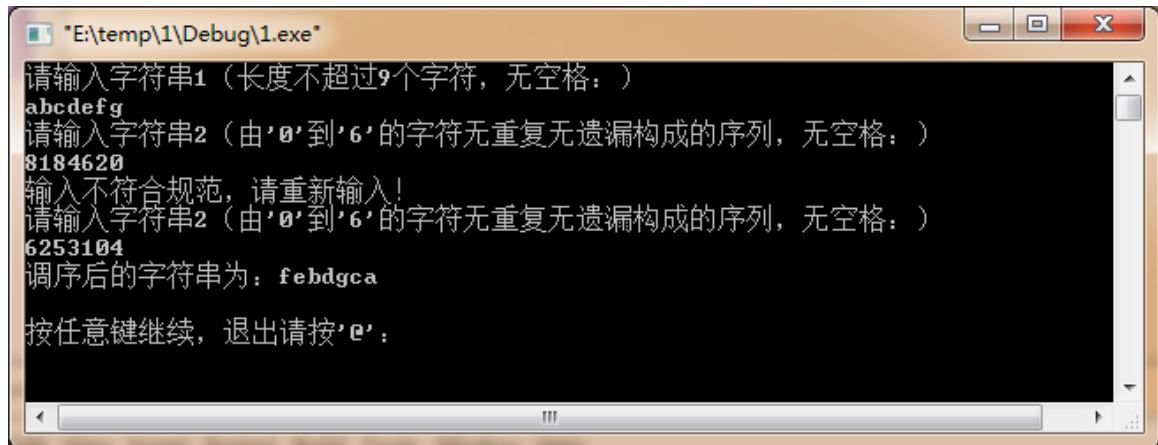
- 主函数 (2 分)**
主函数循环运行，当且仅当输入 '@' 时，程序结束运行。
- 函数 inputStr() (3 分)**
从键盘输入字符串 s (假设，字符串长度不超过 9 个字符，无空格)，并传回给主函数。
- 函数 inputSeq() (10 分)**
从键盘输入一个仅由数字字符构成的字符串 t，要求：
 - 验证字符串 t 的长度与 inputStr() 输入的字符串 s 长度 len 是否相等； (4 分)
 - 验证字符串 t 是否由 0~len-1 这些数字字符无重复、无遗漏构成； (2 分)
 - 如果输入的 t 不符合(1)与(2)的验证要求，则提示重新输入 t，直到 t 复合验证要求，并将 t 返回给主函数。 (4 分)
- 函数 mapping() (15 分)**
 - mapping() 接收从主函数传递过来的字符串 s 和字符串 t，并按照字符串 t 的数字字符顺序对字符串 s 中字符进行重新编号映射，生成新的字符串 new_s (6 分)。
 - 映射的规则为：若字符串 s 中的字符 s[i] 对应的字符串 t 中的字符 t[i] 为 '0'，则字符 s[i] 映射到 new_s 到第 0 位置，若为 '1' 则映射到第 1 位置……依此类推。 (7 分)



- mapping() 将 new_s 传回给主函数并输出至屏幕。 (2 分)

程序运行示例 1:

inputStr 输入 s，inputSeq 输入 t 正确后，则映射后新的字符串 new_s 是：



二、动态矩阵处理。编制程序，实现如下功能：

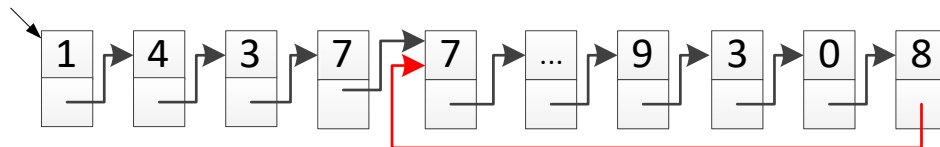
1. **函数 input() (10 分)**
首先从键盘输入整数矩阵的行数 m 与列数 n ，根据输入的 m 与 n ，动态生成整数矩阵 a ，并依次输入矩阵 a 的各元素值，将整数矩阵 a 以及 m 与 n 的值传回给主函数。
2. **函数 RowSort() (8 分)**
对于矩阵 a 的每一行内的各个元素按照从大到小的顺序进行降序排序。
3. **函数 MatrixSort() (9 分)**
该函数逐行调用函数 `RowSort()`，并将矩阵每一行的位置，按照行内所有元素的和的数值，从大到小重排矩阵，最后将排序处理后的矩阵返回给主函数。
4. **函数 print() (4 分)**
按行依次输出 $m \times n$ 的矩阵 a 的各个元素。
5. **编写主函数 main() (4 分)**
按照程序示例，主函数循环运行，当且仅当输入 '@' 时，程序结束运行。

程序运行示例 2:

```
"E:\temp\2\Debug\2.exe"
请输入矩阵的行列值m与n:3 4
请输入矩阵第 0 行各元素的值:
12 10 21 -8
请输入矩阵第 1 行各元素的值:
3 17 6 10
请输入矩阵第 2 行各元素的值:
20 32 7 11
排序处理后的矩阵为:
32 20 11 7
17 10 6 3
21 12 10 -8
请按任意键继续，仅当输入@，程序结束。
```

三、函数编程。按照要求，编写完成函数程序。Void

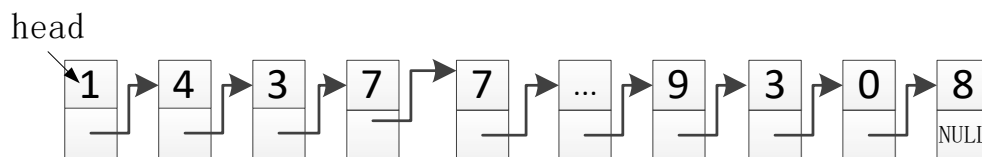
一段可运行的程序如下：其中，函数 `create()` 生成一个长度随机的环链表如下图所示（链表中每一结点数据域也是随机值），该链表最后一个节点的 `next` 指针随机的指向了链表内的某一节点。



随机生成的环链

请编写函数（不允许修改 `create()` 函数和主函数等已有的程序代码）实现以下功能：

1. **解环函数 openloop() (25 分)**
形参是环链表头指针 `head`，`openloop` 函数将链表解环，使之恢复成为一个正常的单链表如下图所示。



解环后的单链表

2. **输出链表函数 print() (10 分)**
输出解环后的单向链表到屏幕。

给定程序:

```
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
```

```

#include <time.h>

struct node
{
    int num;
    struct node *next;
};

struct node *create();
void openloop(struct node *);
void print(struct node *);

void main()
{
    struct node *head;
    srand(time(NULL));
    head = create();
    openloop(head);
    print(head);
}

struct node *create()
{
    int n = 0;
    struct node *p1, *p2, *head;
    int i;
    int randomIndex ;

    head=NULL;

    while (n==0)          //环链的长度随机
        n=((int)rand()) % 20;

    for( i=0; i<n; i++)
    {
        p1 = (struct node *) malloc(sizeof(struct node));
        p1->next = NULL;
        p1->num = ((int)rand()); //每节点内容随机
        if(i == 0)
            head = p1;
        else
            p2->next = p1;
        p2 = p1;
    }

    if (head!=NULL) //当链表不为空时，随机成环
    {
        randomIndex = ((int)rand()) % n; //随机成环的节点
        p2 = head;
        for(i=0; i<randomIndex; i++)
            p2 = p2->next;
        p1->next = p2;
    }

    return head;
}

```

//以上原始代码不允许修改

程序运行示例 3:

```

E:\temp\3\Debug\3.exe
当前链表内容为:
29305 -> 30557 -> 307 -> 12763 -> 11865 -> 25267 -> 2196 -> 19665 -> 8810 -> 12002 -> 17700 -> 25871 -> 23590 -> 13709 -> 25543 -> 1658 -> NULL
Press any key to continue.

```

