2014秋季计算机语言与程序设计

期末试卷

2015 年 1 月 20 日 姓名 班级 学号

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 大题 | 题号 | 一 | 二 | 三 | 四 | 五 | 总分 |
| 分数 | 20 | 30分 | 20 | 20分 | 10分 |
| 成绩 | |  |  |  |  |  |  |

**试卷提交说明**

1. 在考卷上填写你的姓名、学号；
2. 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；
3. 交卷时，提交一个压缩文件包，包括：
   1. 文件包名

学号+姓名，如：2013123456ZhangSan.rar

* 1. 文件包内容
     + - * 每道题的c工程文件夹；
         * 考卷。

1. 除非特别说明，否则不允许使用字符串处理库函数。
2. **字符串处理（本题30分）**

编制一个程序，要求实现如下功能：

1. **主函数（3分）**

主函数循环运行，当且仅当输入’@’时，程序结束运行。

1. **函数inputStrS() （5分）**

从键盘输入字符串str，stra，strb共三个字符串（假设，输入的字符串str长度不超过100个字符，stra、strb字符串的长度都不超过10个字符），并传回给主函数。

1. **函数myStrStr() （10分）**

myStrStr()函数，接受从主函数传递过来的字符串str和字符串stra，并在字符串str中查找stra第一次出现的位置（不包括stra的串结束符），返回该位置的下标值，如果找不到，返回-1。（注，不能使用strstr()库函数来做）

1. **函数replace() （10分）**

**replace()**接收从主函数传递过来的字符串str，stra，strb；通过循环调用myStrStr()函数，找到stra在str中出现的位置，并用字符串strb将stra进行替换，直到str中不再有字符串stra出现。

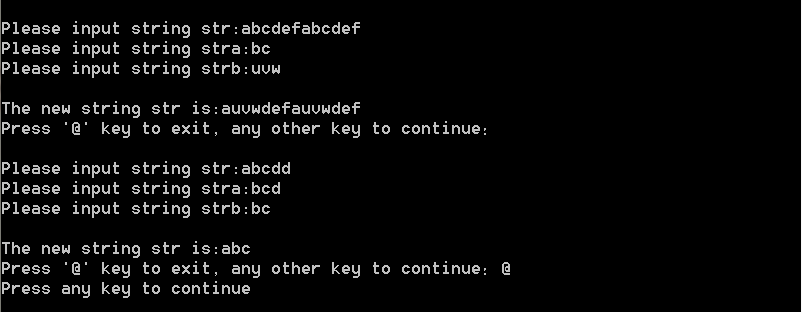
如，输入的字符串str，stra，strb分别为”abcdefabcdef”，”bc”，“uvw”，则最终替换后的字符串变成”auvwdefauvwdef”；

再如，输入的字符串str，stra，strb分别为”abcdd”，”bcd”，“bc”，则最终替换后的字符串变成”abc”。

不考虑，输入的字符串stra同时又是strb子串的情况。如输入的字符串str，stra，strb分别为”abcdd”，”bc”，“bcd”，则无法完成替换。这种情况不考虑。

1. **在主函数中输出替换以后的字符串（2分）**

**程序运行示例1：**



1. **稀疏矩阵处理（本题30分）**

矩阵中，如果多数的元素为0，称此矩阵为稀疏矩阵（sparse matrix）。若一个稀疏整数矩阵a中有k个非零项，那么只需要存储这k个非零项的三元组信息（行号i、列号j、元素值aij）就可以了。

请编制一个程序，实现如下功能：

1. **函数input()** **（10分）**

首先从键盘输入整数矩阵a的行数m与列数n，以及a中的非零项个数k；根据输入的非零项个数k，动态生成存放矩阵a中k个三元组信息的空间；并依次输入矩阵a的各非零元素信息；形成按照行号优先的顺序（即按照先行号、后列号的顺序来存放所有非零元素）存储的整数矩阵a；并将矩阵a以及m、n与k的值传回给主函数。

1. **函数TransposeMatrix ()** **（10分）**

对于按照三元组形式存储的稀疏矩阵a，计算出a的转置矩阵b，并按照行号优先的顺序进行存储；将转置矩阵b以及行数mb、列数nb与非零项个数kb的值传回给主函数。

1. **函数SubMatrix()** **（4分）**

该函数能够从矩阵a中抽取出一个子矩阵c。要求输入a矩阵某个元素的下标(ia，ja)，以及要抽取的c矩阵的行数mc，列数nc；在a矩阵中以(ia，ja)为c的起始顶点，抽取mc行nc列的子矩阵c；并将子矩阵c、以及行数mc、列数nc与非零项个数kc的值传回给主函数。如输入的(ia，ja)、mc、nc不能在a中产生矩阵c，提示重新输入。

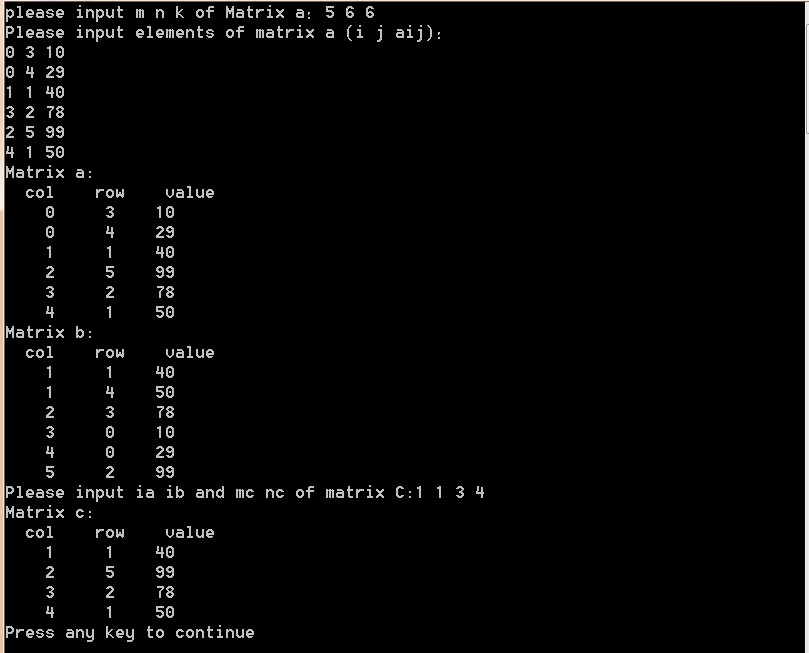
1. **函数print()** **（4分）**

依次输出m\*n的稀疏矩阵a的所有非零元素。

1. **编写主函数main()** **（2分）**

按照程序示例，主函数调用input()函数输入稀疏矩阵a，调用print()函数输出矩阵a；主函数调用TransposeMatrix()函数计算转置矩阵b，调用print()函数输出转置矩阵b；主函数调用SubMatrix()函数抽取子矩阵c，调用print()函数输出子矩阵c。

**程序运行示例2：**



1. **链表编程(本题30分)**

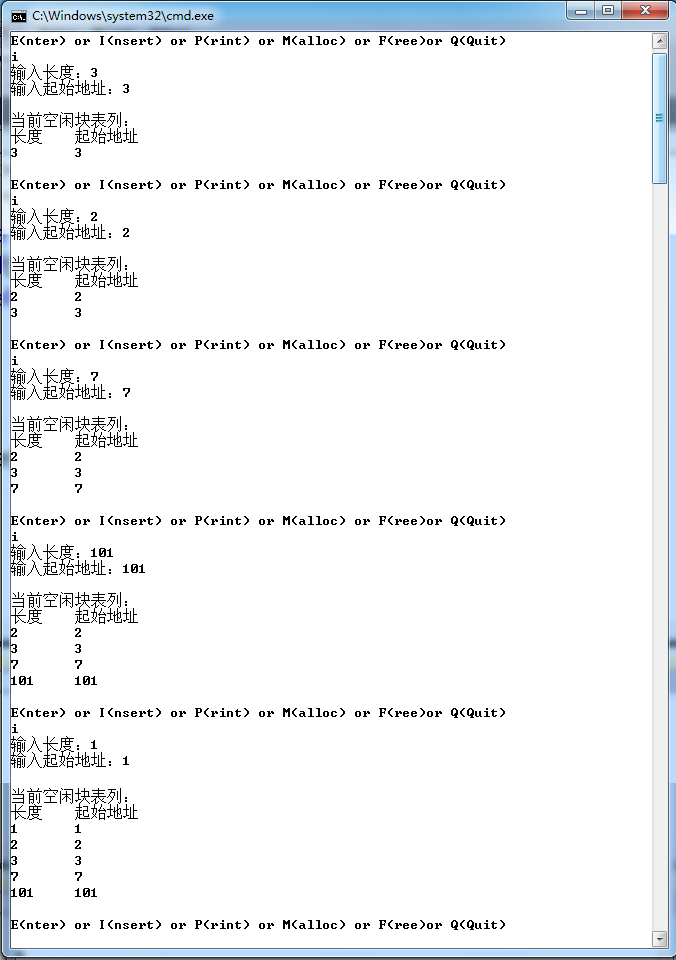
假设你的计算机在某一时刻的内存空闲存储块表列如下：

|  |  |
| --- | --- |
| 长度（kByte）  （int） | 起始地址  （long） |
| 1 | xxxxxx |
| 1 | xxxxxx |
| 3 | xxxxxx |
| 3 | xxxxxx |
| 3 | xxxxxx |
| 3 | xxxxxx |
| 100 | xxxxxx |
| 120 | xxxxxx |
| 130 | xxxxxx |
| 1024 | xxxxxx |
| 1024 | xxxxxx |
| 1024 | xxxxxx |
| …… | ……. |
| 4096 | xxxxxx |
| …… | …… |

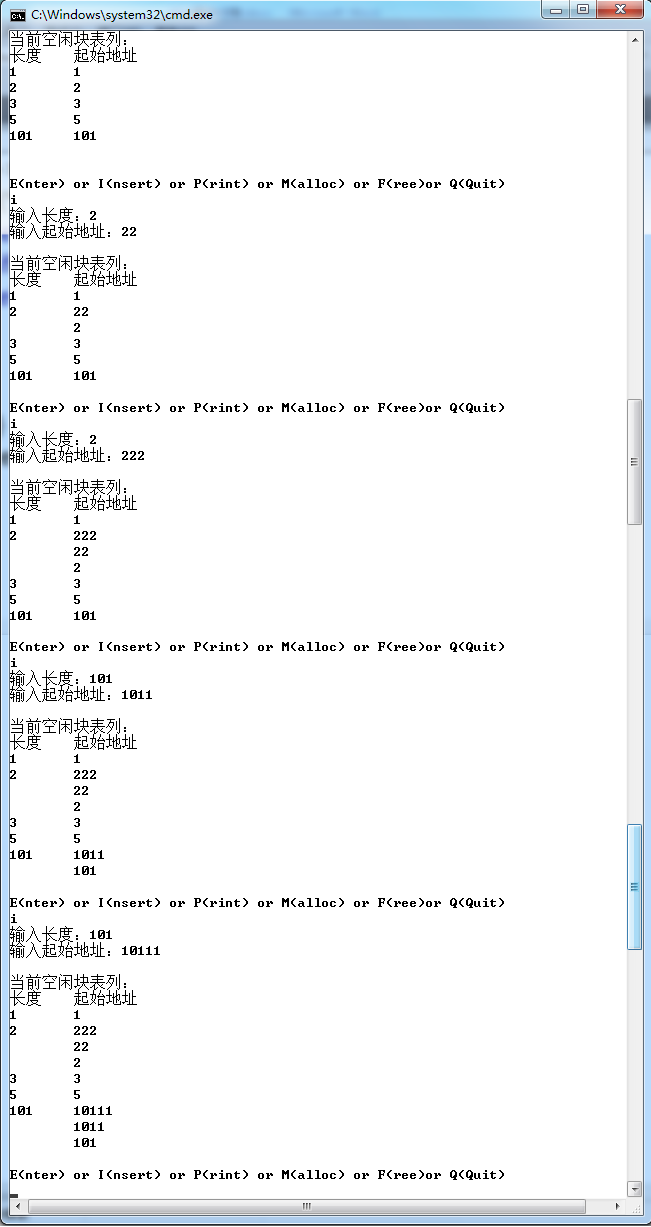
现用一个按存储块长度递增有序的非循环链表管理这些随机插入的空闲存储块，相同长度的存储块数≤Ceiling（比如Ceiling=20），若链表内同长度存储块的节点已经存在，则存储块信息插入到当前节点中。请设计一个内存空闲块管理程序，包括：

1. 实现下述功能的链表节点结构。**（5分）**
2. **输入函数Newenter**：从键盘读入一个空闲块长度和起始地址（地址任意,但不能重复）。**（3分）**
3. **插入函数insert**：用递增有序非循环链存储当前空闲的存储块信息（测试节点数>10，总存储块数>30）。**（5分）**
4. **Mymalloc函数**：用申请的内存长度Length查询链表，等于或大于Length的第一个节点为匹配成功，释放节点内的一个存储块并返回指向该存储块的指针，释放后节点为空的话应从链表中删除此节点，而查询节点不存在则返回空指针提示匹配失败。**（7分）**
5. **Myfree函数**：将Mymalloc返回的指针所指向的存储块（长度和起始地址）交还给链表。**（5分）**
6. **List函数**：遍历当前链表到屏幕。**（5分）**

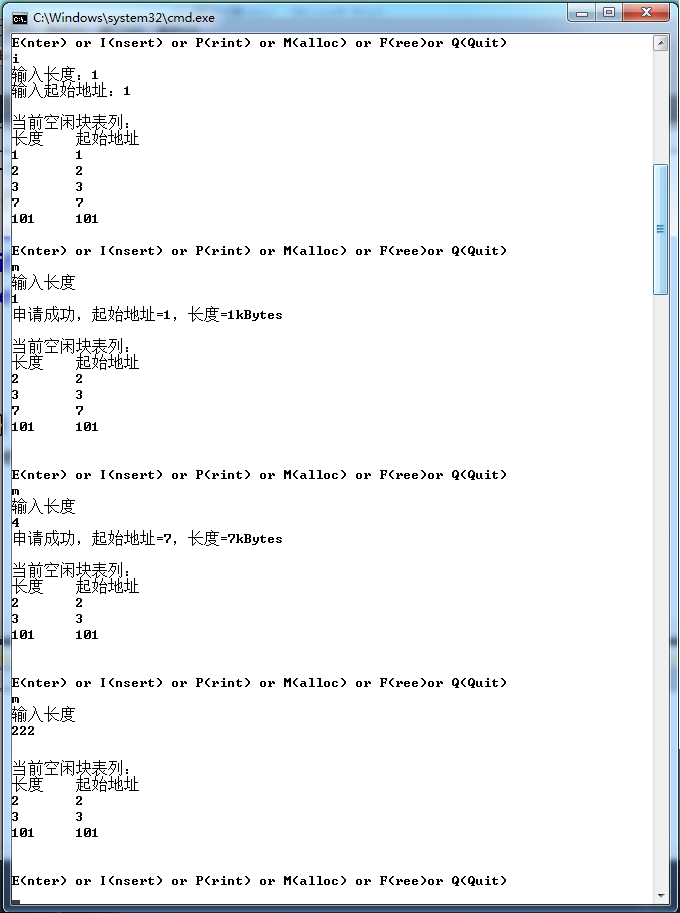
链表插入的测试界面参考如图题3-1a和图题3-1b，节点申请的测试界面参考如图题3-2，节点申请与释放的测试界面参考如图题3-3。



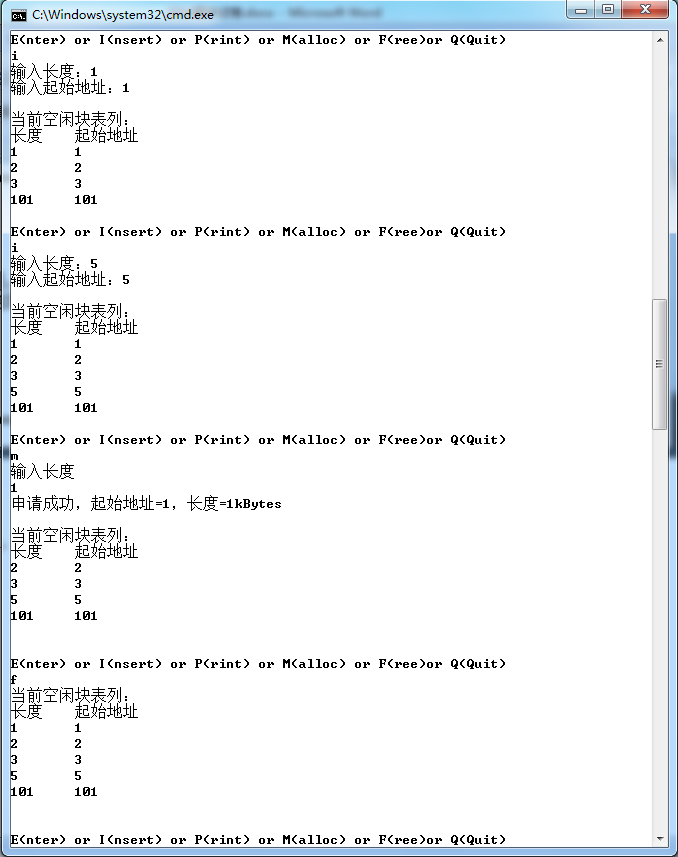
图题3-1a空闲块链表插入



图题3-1b空闲块链表插入



图题3-2节点申请



图题3-3节点申请与释放

1. **递归编程** (本题10分)

给定两个有序字符串str1和str2，这里，有序字符串是指字符串中的字符按ASCII升序排列，例如“abcdefg”，采用递归算法将两个字符串合并成一个新的字符串str3，使得str3也是有序字符串。

输入：两个有序字符串str1和str2，长度不超过20。

输出：合并后的有序字符串str3。

按照要求，编写程序。

1. **函数input()（3分）**

键盘输入str1和str2，并返回给主函数。

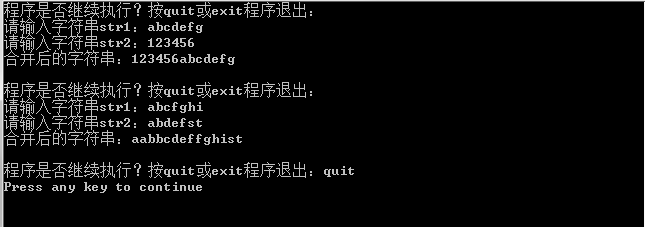
1. **递归函数combine()（5分）**

使用递归的方法，对str1和str2进行合并，得到有序字符串str3。

1. **主函数main()（2分）**

主函数循环运行,当输入“quit”或“exit”时，程序结束运行。否则调用input()输入str1和str2，然后调用递归函数combine()对字符串进行合并，并输出相应的结果。

**程序运行示例4：**



//以上为全部题目。