

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

- 在答卷上填写你的姓名、学号；
- 按要求编程，调试运行，将每道题的结果考屏，粘贴在答卷对应的题后；
- 交卷时，提交一个压缩文件包，包括：
 - 文件包名
学号+姓名，如：2013123456ZhangSan.rar
 - 文件包内容
 - ◇ 每道题的 c 工程文件夹；
 - ◇ 考卷。
- 除非特别说明，否则不允许使用字符串处理库函数。

一、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。

2. 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。

3. 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）

4. 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。

5. 在主函数中输出替换以后的字符串 (2 分)

程序运行示例 1:

```

Please input string str:abcdefabcdef
Please input string stra:bc
Please input string strb:uvw

The new string str is:auvwdefauvwdef
Press '@' key to exit, any other key to continue:

Please input string str:abcdd
Please input string stra:bcd
Please input string strb:bc

The new string str is:abc
Press '@' key to exit, any other key to continue: @
Press any key to continue

```

二、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 a_{ij}) 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 的值传给给主函数。

2. 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。

3. 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。

4. 函数 print() (4 分)

依次输出 $m*n$ 的稀疏矩阵 a 的所有非零元素。

5. 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：

```

please input m n k of Matrix a: 5 6 6
Please input elements of matrix a (i j aij):
0 3 10
0 4 29
1 1 40
3 2 78
2 5 99
4 1 50
Matrix a:
  col   row   value
    0     3     10
    0     4     29
    1     1     40
    2     5     99
    3     2     78
    4     1     50
Matrix b:
  col   row   value
    1     1     40
    1     4     50
    2     3     78
    3     0     10
    4     0     29
    5     2     99
Please input ia ib and mc nc of matrix C:1 1 3 4
Matrix c:
  col   row   value
    1     1     40
    2     5     99
    3     2     78
    4     1     50
Press any key to continue

```

三、链表编程(本题 30 分)

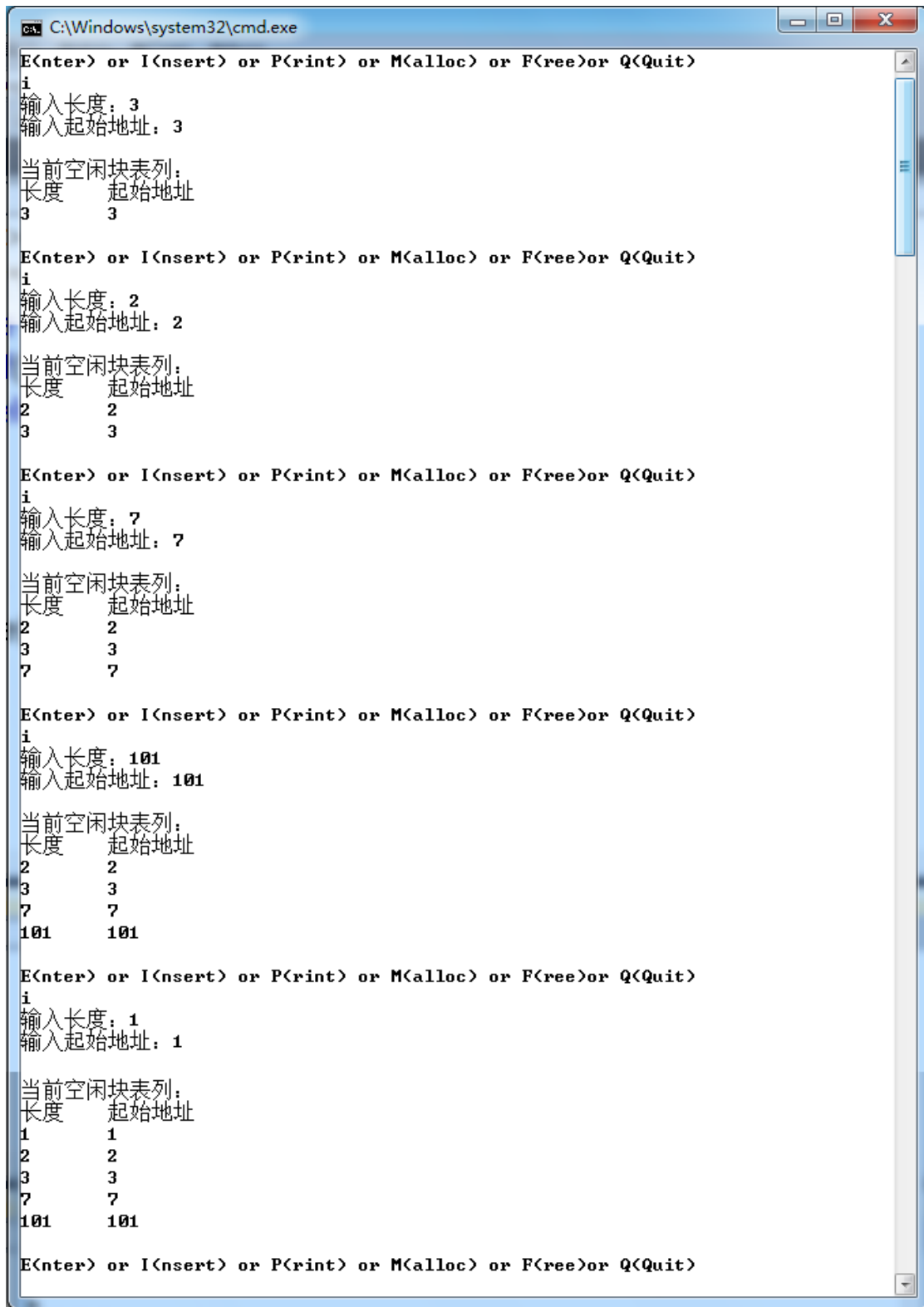
假设你的计算机在某一时刻的内存空闲存储块表列如下：

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
.....

2014 秋季《计算机语言与程序设计基础》期末试卷

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 \leq 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。



```
C:\Windows\system32\cmd.exe
E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(uit)
i
输入长度: 3
输入起始地址: 3

当前空闲块表列:
长度      起始地址
3          3

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(uit)
i
输入长度: 2
输入起始地址: 2

当前空闲块表列:
长度      起始地址
2          2
3          3

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(uit)
i
输入长度: 7
输入起始地址: 7

当前空闲块表列:
长度      起始地址
2          2
3          3
7          7

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(uit)
i
输入长度: 101
输入起始地址: 101

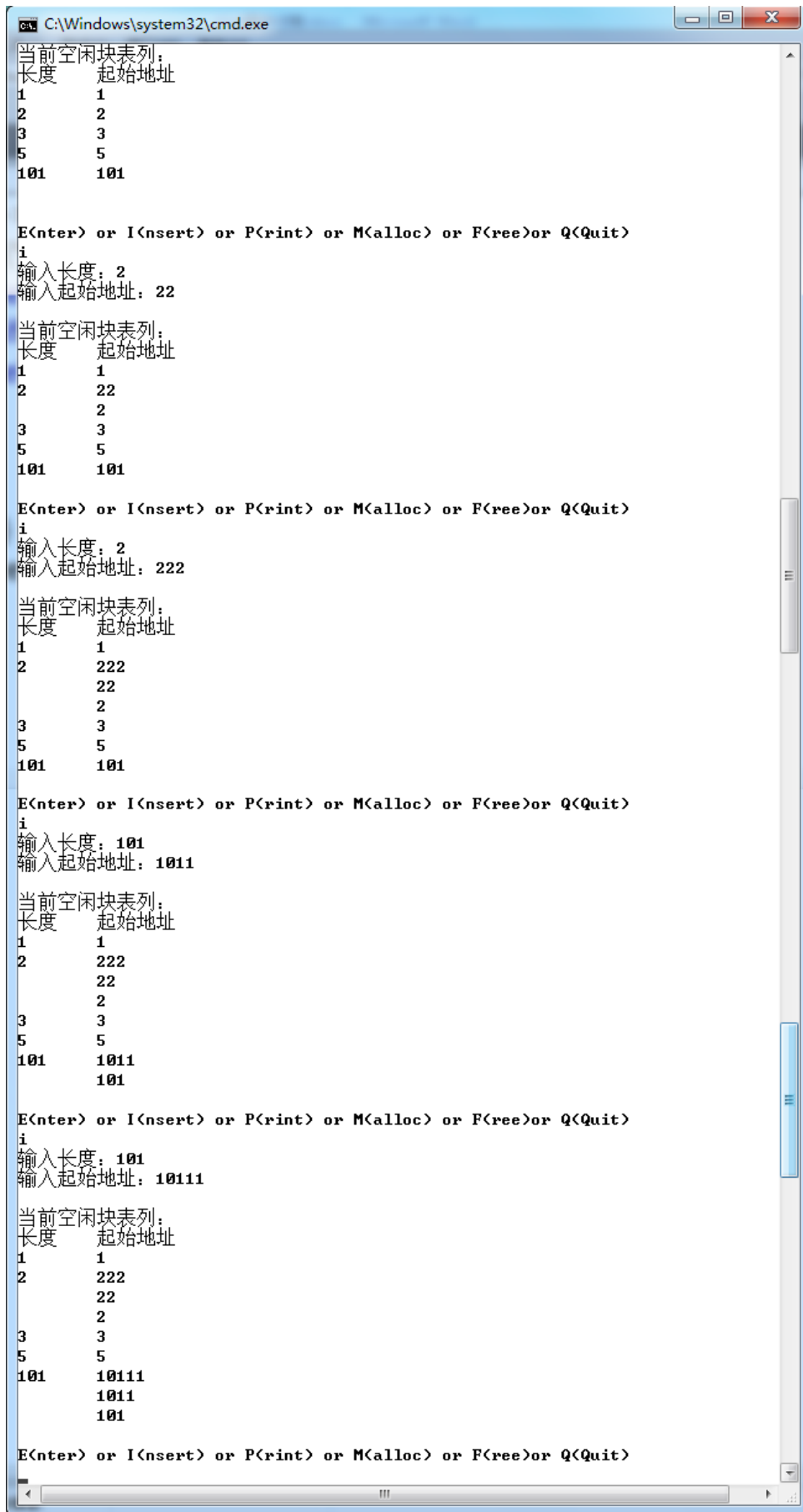
当前空闲块表列:
长度      起始地址
2          2
3          3
7          7
101        101

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(uit)
i
输入长度: 1
输入起始地址: 1

当前空闲块表列:
长度      起始地址
1          1
2          2
3          3
7          7
101        101

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(uit)
```

图题 3-1a 空闲块链表插入



```
C:\Windows\system32\cmd.exe
当前空闲块表列:
长度 起始地址
1      1
2      2
3      3
5      5
101    101

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(uit)
i
输入长度: 2
输入起始地址: 22

当前空闲块表列:
长度 起始地址
1      1
2      22
2      2
3      3
5      5
101    101

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(uit)
i
输入长度: 2
输入起始地址: 222

当前空闲块表列:
长度 起始地址
1      1
2      222
2      22
2      2
3      3
5      5
101    101

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(uit)
i
输入长度: 101
输入起始地址: 1011

当前空闲块表列:
长度 起始地址
1      1
2      222
2      22
2      2
3      3
5      5
101    1011
101    1011
101    101

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(uit)
i
输入长度: 101
输入起始地址: 10111

当前空闲块表列:
长度 起始地址
1      1
2      222
2      22
2      2
3      3
5      5
101    10111
101    1011
101    101

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(uit)
```

图题 3-1b 空闲块链表插入

```
C:\Windows\system32\cmd.exe
E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(Quit)
i
输入长度: 1
输入起始地址: 1

当前空闲块表列:
长度 起始地址
1      1
2      2
3      3
7      7
101    101

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(Quit)
n
输入长度
1
申请成功, 起始地址=1, 长度=1kBytes

当前空闲块表列:
长度 起始地址
2      2
3      3
7      7
101    101

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(Quit)
n
输入长度
4
申请成功, 起始地址=7, 长度=7kBytes

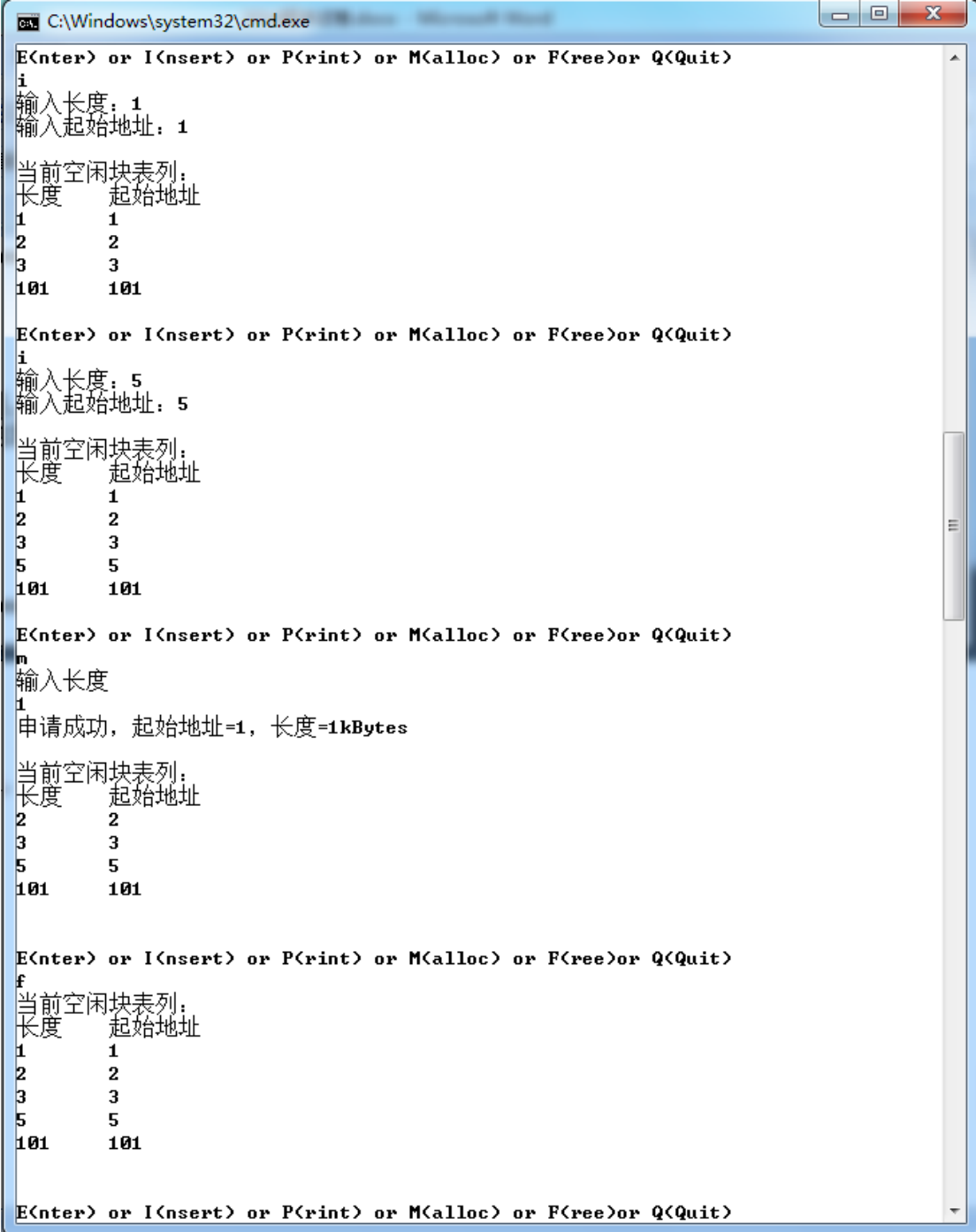
当前空闲块表列:
长度 起始地址
2      2
3      3
101    101

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(Quit)
n
输入长度
222

当前空闲块表列:
长度 起始地址
2      2
3      3
101    101

E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(Quit)
```

图题 3-2 节点申请



```

C:\Windows\system32\cmd.exe
E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(Quit)
i
输入长度: 1
输入起始地址: 1
当前空闲块表列:
长度 起始地址
1      1
2      2
3      3
101    101
E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(Quit)
i
输入长度: 5
输入起始地址: 5
当前空闲块表列:
长度 起始地址
1      1
2      2
3      3
5      5
101    101
E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(Quit)
m
输入长度
1
申请成功, 起始地址=1, 长度=1kBytes
当前空闲块表列:
长度 起始地址
2      2
3      3
5      5
101    101
E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(Quit)
f
当前空闲块表列:
长度 起始地址
1      1
2      2
3      3
5      5
101    101
E(nter) or I(nsert) or P(rint) or M(alloc) or F(ree) or Q(Quit)

```

图题 3-3 节点申请与释放

四、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)

2014 秋季《计算机语言与程序设计基础》期末试卷

键盘输入 str1 和 str2，并返回给主函数。

2. 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。

3. 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:

```
程序是否继续执行? 按quit或exit程序退出:
请输入字符串str1: abcdefg
请输入字符串str2: 123456
合并后的字符串: 123456abcdefg

程序是否继续执行? 按quit或exit程序退出:
请输入字符串str1: abcfghi
请输入字符串str2: abdefst
合并后的字符串: aabbcdfeffghist

程序是否继续执行? 按quit或exit程序退出: quit
Press any key to continue
```

//以上为全部题目。