

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） 2020 年 1 月 7 日

姓名_____ 班级 96 学号_____

大题	题号	一	二	三	四	总分
	分数	30 分	30 分	33 分	7 分	
成绩						

试卷提交说明

1. 在考卷上填写你的姓名、学号；
2. **所有程序，都只能使用 C89/C99 标准包含的头文件；**
3. 按要求编程，调试运行，将每道题的结果拷屏，粘贴在考卷对应的题后；
4. 交卷时，提交一个压缩包，包括：
 - a) 文件包名“学号.zip”，如 2019123456.zip
 - b) ZIP 文件包内容
 - ✧ 每道题的 C 语言源文件，文件名为“学号_题号.c”，如 2019123456_1.c；
 - ✧ 考卷，文件名为“学号_考卷.docx”
 - ✧ 例如：2019123456.zip 应包括 2019123456_1.c、2019123456_2.c、2019123456_3.c、2019123456_4.c、2019123456_考卷.docx，共 5 个文件。

一、字符串处理（30 分）

请按照要求编写程序，程序结构如下：

1. 函数 Input()（10 分）

该函数的功能是从键盘输入两个字符串 str1、str2，以及两个整数 g、k，并能将上述输入的内容传回主调函数。该函数具体的要求为：（1）假设用户每次输入的 str1、str2 两个字符串长度都介于 2 到 100 个字符之间，并且输入的字符串中的可能包含空格。（2）要求输入的整数 $0 \leq g < \text{字符串 str1 的长度}$ ， $1 < k \leq \text{字符串 str2 的长度}$ ，若 g、k 不满足上述条件，提示重新输入这两个整数。

2. 函数 Combine()（10 分）

通过参数接收主函数提供的字符串 str1、str2 和整数 g，实现对字符串 str1 和 str2 的合并操作，得到新的字符串 str3，并能将 str3 返回给主调函数。合并操作的要求是将字符串 str2 插入到在字符串 str1 的第 g 个字符之后的位置，字符位置计数从 0 开始。如字符串 str1、str2、g 分别是“abcdefg”、“uvw”、2，则得到字符串 str3 为“abcuvwdefg”。

3. 函数 KReverse()（5 分）

通过参数接收主函数提供的字符串 str3 和整数 k，实现对字符串 str3 的一种循环局部逆序操作，并能将更新后的字符串 str3 返回给主调函数。逆序操作的要求是，从字符串 str3 的首字符开始，每 k 个字符作为一个子段，并在每个子段内进行逆序操作，最后不足 k 个字符的剩余部分不需做逆序操作。如字符串 str3、整数 k 分别是“abcuvwdefg”、3，则字符串 str3 循环局部逆序后变为“cbawvufedg”。

4. 主函数 main()（5 分）

- （1）定义所有使用到的字符串及各变量。
- （2）调用 Input() 函数，完成输入操作。
- （3）调用 Combine() 函数，完成 str1 与 str2 的连接；在主函数输出 str3。
- （4）调用 KReverse() 函数，完成字符串 str3 的循环局部逆序处理；在主函数输出 str3。

[程序示例]

```

请输入字符串str1（最大长度不超过100）： abcdefg
请输入字符串str2（最大长度不超过100）： uvw
请输入整数g（0≤g<字符串str1的长度）： 8
请重新输入整数g（0≤g<字符串str1的长度）： 2
请输入整数k（0<k≤字符串str2的长度）： -1
请重新输入整数k（0<k≤字符串str2的长度）： 3

按位置插入合并后的字符串str3为： abcuvwdefg
循环局部逆序后的字符串str3为： cbawvufedg

请按任意键继续. . .

```

请在此处粘贴运行结果截图

二、数组处理（30 分）

编制一个程序，要求实现如下功能：

1. 函数 InputMatrix()（15 分）

该函数提示用户从键盘输入两个大于 0 的整数 m 与 n ，根据输入的整数值动态生成两个整数矩阵 **MatrixA**（ m 行 n 列）、**MatrixB**（ n 行 m 列），并提示用户分别输入两个矩阵的各元素值，然后将上述输入的内容传给主调函数。假设用户输入的 m 、 n 等值是符合要求的，该函数不用判断。

2. 函数 InputPosition()（5 分）

从键盘输入两个非负整数 g 、 k ，要求 $0 \leq g < m$ ， $0 \leq k < n$ ，若输入的 g 、 k 值不符合上述的区间，提示重新输入，并将输入的值传回主函数。

3. 函数 MatrixMerge()（5 分）

根据参数输入，以矩阵 **MatrixA** 的元素 (g,k) 为左上定点位置，将矩阵 **MatrixB** 在此处与 **MatrixA** 进行合并，得到新的矩阵 **MatrixC**，将动态生成的矩阵 **MatrixC** 返回给主调函数。

合并的规则为，在将 **MatrixB** 与 **MatrixA** 进行合并时，若与 **MatrixC** 对应位置的两个矩阵都有元素则将它们相加作为 **MatrixC** 的元素值，若 **MatrixC** 对应的位置只有一个矩阵含有元素值则保留这个矩阵的元素值，若 **MatrixC** 对应的位置没有 **MatrixA** 和 **MatrixB** 的元素，则 **MatrixC** 的元素值为 0。

$$\text{如： } A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \end{pmatrix}, B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \end{pmatrix}, \text{位置 } (g,k) \text{ 为 } (1, 1),$$

$$\text{则矩阵 } C = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7+1 & 8+2 & 9+3 & 10 \\ 11 & 12+4 & 13+5 & 14+6 & 15 \\ 0 & 7 & 8 & 9 & 0 \\ 0 & 10 & 11 & 12 & 0 \\ 0 & 13 & 14 & 15 & 0 \end{pmatrix}, \text{为了清晰起见，两个矩阵重叠部分的}$$

元素写的是表达式。再如，若矩阵 A 、 B 仍为上例中的矩阵，而位置 (g,k) 为 $(2, 3)$ ，

$$\text{则新矩阵 } C = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 0 \\ 6 & 7 & 8 & 9 & 10 & 0 \\ 11 & 12 & 13 & 14+1 & 15+2 & 0+3 \\ 0 & 0 & 0 & 0+4 & 0+5 & 0+6 \\ 0 & 0 & 0 & 0+7 & 0+8 & 0+9 \\ 0 & 0 & 0 & 0+10 & 0+11 & 0+12 \\ 0 & 0 & 0 & 0+13 & 0+14 & 0+15 \end{pmatrix}$$

4. 编写主函数 main()（5 分）

- (1) 定义所使用到的变量。
- (2) 依次调用 `InputMatrix()`、`InputPosition()`函数完成输入。
- (3) 调用 `MatrixMerge()`函数，完成新数组 `MatrixC` 的计算。
- (4) 在主函数中清晰输出数组 `MatrixC` 的各元素值。

[程序示例]

```
请输入两个表示矩阵大小的正整数m、n: 3 5
请按行输入矩阵MatrixA的各元素 (3*5):
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15
请按行输入矩阵MatrixB的各元素 (5*3):
1 2 3
4 5 6
7 8 9
10 11 12
13 14 15

请输入两个表示合并位置的非负整数g、k: 4 1
错误, 请输入两个表示合并位置的非负整数g、k: 1 1

新生成矩阵MatrixC (6*5) 的各元素依次为:
1 2 3 4 5
6 8 10 12 10
11 16 18 20 15
0 7 8 9 0
0 10 11 12 0
0 13 14 15 0

请按任意键继续. . .
```

请在此处粘贴运行结果截图

三、链表编程 (33 分)

一位同学的信息包含了如下的内容: 学号 (长度为 4 的字符串)、班级 (字符)、身高 (整数)。请按照以下要求, 编写完成链表程序。

1. 节点类型 `student` 定义 (3 分)

2. 函数 `MyCreate()` (10 分)

该函数完成如下的功能

- 从键盘依次输入若干个同学的信息 (包含学号、班级、身高信息, 并假设输入的学号信息都未重复, 不用判断)。
- 按照键盘输入的顺序, 构造一个单链表。
- 若输入的学号信息为 “0000”, 则表示输入结束。
- 函数返回单链表首节点指针。

3. 函数 `MyPrint()` (5 分)

在屏幕上依次清晰输出链表中的各节点信息。

4. 函数 `MyFree()` (5 分)

释放链表的各节点空间。

5. 函数 `MySort()` (4 分)

对于上述函数创建的链表进行重新排序的操作, 排序的规则是: (1) 按照班级从低到高进行排列; (2) 若班级相同则按照身高从低到高进行排序; (3) 若班级和身高均相同, 则按照学号从低到高的顺序进行。将顺序调整好的链表通过函数返回值传回主调函数。

6. 函数 `MySearch()` (4 分)

接收主调函数的班级 (字符)、逆序位次 `k` (整数) 等参数信息, 在上述排好顺序的链表中, 查找该班级内逆序位次为 `k` 的同学信息, 并输出, 若无该班级或者该班级中人数不足 `k` 人, 则输出未找到的提示信息。逆序位次的定义为同一班级中按照上述排序规则排序后, 从后往前数第 `k` 个位置的节点。

7. 主函数 `main()` (2 分)

依次调用上述的函数，完成相应功能。

- (1) 调用 `MyCreate()`，生成链表；调用 `MyPrint()`，输出当前链表。
- (2) 调用 `MySort()`，排列好链表顺序；调用 `MyPrint()`，输出当前链表。
- (3) 提示输入班级字符和逆序位次整数；调用 `MySearch()`，输出查找结果。
- (4) 调用 `MyFree()`函数，释放动态生成的链表所有节点空间。

[程序示例]

```
请输入同学信息（学号为0000结束）：
0001 A 175
0002 B 180
0003 C 165
0004 B 178
0005 A 165
0006 C 177
0007 B 175
0008 B 173
0009 A 181
0010 B 180
0011 C 172
0012 B 180
0000 A 0

创建后的链表信息为：
(0001 A 175)->(0002 B 180)->(0003 C 165)->(0004 B 178)->(0005 A 165)->(0006 C 177)->(0007 B 175)->(0008 B 173)->(0009 A 181)->(0010 B 180)->(0011 C 172)->(0012 B 180)

排序后的链表信息为：
(0005 A 165)->(0001 A 175)->(0009 A 181)->(0008 B 173)->(0007 B 175)->(0004 B 178)->(0002 B 180)->(0010 B 180)->(0012 B 180)->(0003 C 165)->(0011 C 172)->(0006 C 177)

请输入待查的班级和逆序位次信息：B 3
该同学的信息为(0002 B 180)

请按任意键继续. . .
```

输入样例数据

```
0001 A 175
0002 B 180
0003 C 165
0004 B 178
0005 A 165
0006 C 177
0007 B 175
0008 B 173
0009 A 181
0010 B 180
0011 C 172
0012 B 180
0000 A 0
```

请在此处粘贴运行结果截图

四、递归编程（7分）

老师有一些书籍需要在书架上从左到右依次排列，若老师的书籍数目最大不超过 10 本，每本书籍的名称不超过 15 个字节，考虑名称相同的书籍算做同一种书籍的情况。请使用递归编程的方法，将所有可能的书籍排列顺序在屏幕上输出。

编程要求如下：

1. 函数 `Input()`（1分）

提示从键盘输入书籍的个数，然后依次输入书籍的名称。（假设用户输入的信息符合要求，不用判断）

2. 递归函数 `PlacingBooks()`（5分）

考虑名称相同的书籍算做同一种书籍，将所有可能的书籍摆放顺序在屏幕上输出（输出格式为每种摆法为 1 行，书籍的名字占用宽度为 15 个字节且右对齐）。

3. 主函数 `main()`（1分）

调用上述函数，实现程序功能。

[程序示例]

```
请输入书籍的个数 (<=10): 4
请输入书籍的名称 (4 个):
C Programming
Python
C Programming
Java

这些书籍的摆法如下:
C Programming      Python  C Programming      Java
C Programming      Python      Java  C Programming
C Programming  C Programming      Python      Java
C Programming  C Programming      Java      Python
C Programming      Java  C Programming      Python
C Programming      Java      Python  C Programming
      Python  C Programming  C Programming      Java
      Python  C Programming      Java  C Programming
      Java      Python  C Programming  C Programming
      Java  C Programming      Python  C Programming
      Java  C Programming  C Programming      Python

请按任意键继续. . .
```

请在此处粘贴运行结果截图

//以上为全部题目。