2016 年《程序设计导论》课程期末测试

2016—2017 学年第 1 学期

班级 _____ **姓名** _____ **学号**_____

注意事项

1. 本次测试的时间为 180 分钟；编程结果采用机器自动评测。
2. 共有 6 题，第 1、2 题，25 分；第 5、6 题，15 分；第 3、4 题任选其一，20 分。
3. 提交到在线评测系统中的程序均采用标准输入和标准输出（键盘输入和屏幕输出）。
4. 程序设计语言选用 C 或 C++。
5. 所有题目的时间限制均为 1s。

一、进制转换（25 分）

得分	评卷人

【问题描述】

编程实现，将十进制整数 d，转换为 h 进制数 a，并按 h 进制规则输出 a。

【输入格式】

一行，两个十进制整数 d 和 h，之间由一个空格分隔，其中：0≤d≤1,000,000,000；2≤h≤16。

【输出格式】

一个 h 进制的整数。十进制以上数字，用大写字母 A、B、C、D、E、F 编码。

【输入样例】

123 16

【输出样例】

7B

二、字符串移位包含（25 分）

得分	评卷人

【问题描述】

给定两个字符串 s1 和 s2，要求判定 s2 是否能够被通过 s1 作向右循环移位（rotate）得到的字符串包含。例如，给定 s1 为 AABCD 和 s2 为 CDAA，返回 true；给定 s1 为 ABCD 和 s2 为 ACBD，返回 false。注：右循环移位是指将字符串中每个字符向右侧移动一个位置，最右侧的字符移动到最左侧。例如字符串 ABC 做一次循环移位是 CAB。

【输入格式】

输入为 2 行，依次为 s1 和 s2 字符串。两个字符串的长度都不超过 26，字符为大写英文字母。

【输出格式】

分别为 true 或 false。

【输入样例 1】

AABCD
CDAA

【输出样例 1】

true

【输入样例 2】

ABCD
ACBD

【输出样例 2】

false

三、平均成绩排序（20 分）

得分	评卷人

【问题描述】

有 n 位学生，每位学生修读的科目数不尽相同，已知所有学生的各科成绩，要求按学生平均成绩由高到低输出学生的学号、平均成绩；当平均成绩同时，按学号从低到高排序。对平均成绩，只取小数点后前 2 位，从第 3 位开始**舍弃**（无需舍入）。

【输入格式】

输入为 n+1 行，第一行为 n 表示学生人数。

从第二行开始的 n 行，每行为一名学生的成绩信息，包括：学号、科目数，各科成绩。其中 n、学号、成绩均为整数，它们的值域为：

0≤n≤10000，1≤学号≤1000000，0≤成绩≤100。学生的科目数都不超过 100 门。

【输出格式】

最多 n 行，每行两个数，学号在前，后为平均成绩，空格分隔。若 n 为 0，输出 NO；若某学生所修科目不到 2 门，则不纳入排序，若无人修满 2 门，也输出 NO。

【输入样例】

```
5
1001 2 89 78
2003 4 88 99 100 88
4004 3 72 80 61
1004 3 70 61 82
3001 1 100
```

【输出样例】

```
2003 93.75
1001 83.50
1004 72.66
4004 72.66
```


学号_____

四、整数乘法（20 分）

得分	评卷人

【问题描述】

有两个非负整数 n 、m，但其中一个的数值可能非常巨大，编程精确计算它们相乘的结果。两个数中，大的数位可能高达 200 位，小的一个则不超过 5 位。

【输入格式】

一行，空格分隔的两个整数。哪个大哪个小，并不确定。

【输出格式】

一行，两个整数相乘后的结果。

【输入样例 1】
1000 17176616161616164646

【输出样例 1】
1717661616161616464600**0**

【输出样例 2】
77666767767667 0

【输出样例 2】
0

【数据规模和约定】

40%的数据，乘积在整数范围内；20%在长长整型范围内；40%巨大，不在任何整型范围内。

五、黄页建立（15 分）

得分	评卷人

【问题描述】

给定 N 个人的姓名和电话号码，要求按照姓名字典序建立这些人的黄页。如两个人姓名相同，则按照出现（输入时自然的）顺序建立黄页。

注意：本题为部分代码提交题，系统已经写好主函数和输出函数（display），用户需要**自定义结构体，**并填写 create 函数实现黄页的建立。

【输入格式】

第一行一个正整数 N (0<N<=1000)。

接下来 N 行，每行一个字符串（仅包含小写字母，长度不超过 20 个字符）和一个数字（int 范围内），中间用空格隔开，分别代表姓名和电话号码。

【输出格式】

N 行，每行一个字符串和一个数字代表建立后的黄页。 注意：本题已有代码已实现了数据的输出。

【输入样例】
5
bob 472
can 321
alice 123
alice 321
alice 331

【输出样例】
alice 123
alice 321
alice 331
bob 472
can 321

【代码结构】

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

void display(YellowPage *head)
{
   YellowPage * node = head;
   printf("display data\n");
   while(node != NULL)
   {
      printf("%s %d\n", node->name, node->telNum);
      head = node->next;
   }
}

int main()
{
   display(create());
   return 0;
}
```

【代码需求】

1）根据已有代码，完成单向链表节点类型的定义；

2）定义 create()函数，实现数据读入，链表创建，返回头节点指针。

学号_____

六、DNA 模糊匹配（15 分）

得分	评卷人

【题目描述】

美国联邦调查局(FBI)于 2000 年成立了联邦 DNA 数据库单元 FDDU，收录了重刑犯和少数其他罪犯的法证 DNA。虽然该数据库中的 DNA 数据不一定能和犯罪现场提取的 DNA 完全吻合，但却有可能因为足够的相似性，而将嫌疑锁定到与某名罪犯有血缘关系的家人身上。请你来写一段程序，帮助 FBI 判断两段 DNA 是否足够相似。

DNA 的全称是脱氧核糖核酸，由以下四种碱基组成的双螺旋结构：A(ADENINE 腺嘌呤)、T(THYMINE 胸腺嘧啶)、G(GUANINE 鸟嘌呤)、C(CYTOSINE 胞嘧啶)。因此，任何一段 DNA 都是有 A、C、T、G 四种碱基组成的碱基串。

判断两段 DNA 是否相似有很多标准，这里我们考虑被广泛使用的“编辑距离”函数，该函数的思想是度量将一段 DNA 变成另一个段 DNA 所需要的最少操作数。它首先定义了将一段 DNA 变为另一段 DNA 的三种基本编辑操作：

- **插入**：在任意位置插入一个碱基，如将 ACTG 变为 ACTCG 只需在前者的第三个碱基后面插入一个碱基 C
- **删除**：删除任意一个碱基，如将 ACTG 变为 ATG 只需删除前者的第二个碱基
- **替换**：替换任意一个碱基，如将 ACTG 变为 ACTC 只需将前者的第四个碱基由 G 替换为 C

将一段 DNA 变成另一段 DNA 通常需要上述操作的一个序列。例如，将 ACTG 变为 GTAG 需要：1) 将第一个 A 替换成 G（替换操作）；2) 将第二个 C 删除（删除操作）；3)在第二个碱基后面插入碱基 A（插入操作），编辑操作次数为 3 。不难想象，还会有其它的操作序列也可以将 ACTG 变为 GTAG，每个序列都会有不同的编辑操作次数。

现给定两个碱基串，请你判断二者是否能够匹配。

【输入格式】

输入只有一行，包含两部分，中间用空格分隔：

1) 第一段 DNA 序列
2) 第二段 DNA 序列

数据约定：每个 DNA 序列的长度都不超过 300。

【输出格式】

输出包含一行，包括两部分，中间用空格分隔：

1) 两段 DNA 的编辑距离值，为一个整数；
2) 两段 DNA 是否匹配。判断匹配的标准是，编辑距离值是否小于等于最短 DNA 长度的一半，如果是则输出 “matched”，否则输出“unmatched”。

【输入样例 1】

ACTGGA ACGG

【输出样例 1】

2 matched

【输入样例 2】

ACTGATTGACTGATTGACTGAT TGACTGTTTGACTG

【输出样例 2】

9 unmatched

【解题思路提示】

核心思路：将复杂的问题分解成相似的子问题

假设 DNA 段 a 共包含 m 个碱基：记为从 a[1] 到 a[m]；DNA 段 b 共包含 n 个碱基，从 b[1] 到 b[n]。我们用 d[i][j] 表示将碱基串 a[1]-a[i]转换为碱基 b[1]-b[j] 的编辑距离。

那么有如下规律，下面用 a[i] 和 b[j] 分别表示 a 和 b 的最后一位：

- 当 a[i] 等于 b[j] 时，可知编辑距离 d[i][j] = d[i-1][j-1]，即等于 a 和 b 分别去掉最后一位的编辑距离；
- 当 a[i] 不等于 b[j] 时，计算 d[i][j] 是如下 3 项编辑操作的最小值：
  - d[i-1][j] + 1（删除 a[i]）
  - d[i][j-1] + 1（插入 b[j]）
  - d[i-1][j-1] + 1（将 a[i] 替换为 b[j]）

以上规律存在边界条件：

- a[i][0] = i, b 串为空，表示将 a[1]-a[i] 全部删除，所以编辑距离为 i
- a[0][j] = j, a 串为空，表示 a 插入 b[1]-b[j]，所以编辑距离为 j

请你根据以上规律写出程序，需要特别考虑如何避免重复计算。

学号_____