

2017 清华大学优秀大学生暑期夏令营

THUSCU 2017

第一试

时间：2017 年 7 月 7 日 17:00 ~ 21:30

| | | | |
|---------|--------|--------|--------|
| 题目名称 | 矩阵 | 工资 | 偏差 |
| 题目类型 | 传统型 | 传统型 | 传统型 |
| 输入 | 标准输入 | 标准输入 | 标准输入 |
| 输出 | 标准输出 | 标准输出 | 标准输出 |
| 每个测试点时限 | 1.0 秒 | 1.0 秒 | 1.0 秒 |
| 内存限制 | 512 MB | 512 MB | 512 MB |
| 测试点数目 | 10 | 10 | 10 |
| 每个测试点分值 | 10 | 10 | 10 |

矩阵 (matrix)

【题目背景】

矩阵是高等代数学中的常见工具，也常见于统计分析等应用数学学科中。在物理学中，矩阵于电路学、力学、光学和量子物理中都有应用；计算机科学中，三维动画制作也需要用到矩阵。矩阵的运算是数值分析领域的重要问题。

【题目描述】

你需要维护一个元素均为正整数的 n 阶矩阵，并支持以下几种操作：

1. 将该矩阵转置，即沿主对角线翻转。
2. 将该矩阵行循环移动 k 次，即第 1 行移动至第 $k+1$ 行，第 2 行移动至第 $k+2$ 行……，第 $n-k+1$ 行移动至第 1 行……，第 n 行移动至第 k 行。
3. 将该矩阵列循环移动 k 次，即第 1 列移动至第 $k+1$ 列，第 2 列移动至第 $k+2$ 列……，第 $n-k+1$ 列移动至第 1 列……，第 n 列移动至第 k 列。

【输入格式】

从标准输入读入数据。

第一行一个正整数 n 表示矩阵大小。

随后 n 行每行 n 个空格隔开的整数，表示矩阵中的元素。

随后一行一个正整数 q 表示操作次数。

随后 q 行每行表示一个操作，形如 1 或 2 k 或 3 k 。

【输出格式】

输出到标准输出。

对于每组数据，输出 n 行每行 n 个空格隔开的整数，表示依次进行所有操作后矩阵中的元素。

【样例 1 输入】

```
2
1 2
3 4
2
1
2 1
```

【样例 1 输出】

2 4

1 3

【样例 2】

见题目目录下的 *2.in* 与 *2.ans*。

【子任务】

对于 30% 的数据，保证 $n, q \leq 10$ 。

对于另外 20% 的数据，保证只有操作 1。

对于 100% 的数据，保证 $1 \leq n, q \leq 100, 0 \leq k < n$, 矩阵中元素的值不超过 10^9 。

工资 (wage)

【题目背景】

wangyurzee7 和 Yazid 是好朋友。

【题目描述】

wangyurzee7 是一个繁忙的工作者，他每天都要写很多有趣的代码。

Yazid 开了一家有趣的公司，wangyurzee7 当上了有趣的公司的项目经理。这家公司给码农们结算工资的方式也十分有趣，每个人的薪水正比于所写代码的长度。不过很快，有趣的码农们学会了使用长长的变量名、大括号换行、增加长长的注释、tab 换成空格等等方法，这让 Yazid 头疼不已。

有一天，wangyurzee7 收到了 Yazid 的一个任务：在这个月结算工资之前，写一个脚本把程序中所有的注释全部去掉，再将所有的函数名、变量名替换成单个字母，然后再进行统计代码长度。

这个任务非常简单，但 wangyurzee7 实在是太弱了，他只在 github 上找到了批量替换函数名的轮子，而没有找到去掉注释和替换变量名的轮子，而且这个代码看起来不是那么有趣，wangyurzee7 写不动。

Yazid 勃然大怒，他根本没想到这么简单的任务 wangyurzee7 都不能胜任。他把 wangyurzee7 狠狠地批判了一番。

当然啦，责任还是要由 wangyurzee7 来承担。无奈之下，他只好退而求其次，只需要针对 C/C++ 代码，去掉里面的注释就向 Yazid 交差：

1. 单行注释 `//`：一旦出现，并且被 C/C++ 编译器认为是注释，那么从 `//` 符号开始到行末都被算作注释，比如下面程序的 `//233` 就会被算作注释：

```
#include <stdio>
int main()
{
    printf("//"); //233
    return 0;
}
```

并且如果单行注释最末尾的可见字符为 `\` 字符，则下一行从行初到行末均被认为是注释。

2. 多行注释 `/* */`：一旦出现，并且被 C/C++ 编译器认为是注释，那么从 `/*` 符号开始到 `*/` 符号结束都被算作注释，比如下面程序的 1~4 行就会被算作注释：

```

/* Welcome to THUSCU2017!
 * This is the easiest problem!
 * \\\\\\puts("\\\\\\"**\\");\\\\\\
 *///
int main(**/**/){return!printf("%d",'//');}

```

3. 注释的处理方法：对于每个注释，将从注释的开始一直到注释的结束的字符串，替换成一个空格，其他地方的代码不允许做任何改动。将处理完的代码输出即可。比如上面的两段代码会变成下面这样（第一份代码第四行最末尾有一个空格，第二份代码第一行有两个空格）：

```

-----
#include <stdio>
int main()
{
    printf("//");
    return 0;
}
-----

int main( ){return!printf("%d",'//');}
-----

```

既然要做的东西变少了，问题也就变得更简单了。请你帮帮可怜的 wangyurzee7 解决这个问题。

【输入格式】

从标准输入读入数据。

输入有若干行，每一个测试样例都是一个能够在 C/C++ 编译器下编译运行的代码，里面包含了若干注释。

【输出格式】

输出到标准输出。

输出经过删除注释，并在原位置替换空格过后的 C/C++ 代码。

【样例 1】

见题目目录下的 *1.in* 与 *1.ans*。

【样例 2】

见题目目录下的 *2.in* 与 *2.ans*。

【样例 3】

见题目目录下的 *3.in* 与 *3.ans*。

【子任务】

对于 30% 的数据，保证只有单行注释 `//`，但是并不保证不跨行。

对于另外 40% 的数据，保证只有多行注释 `/* */`，但是并不保证不跨行。

存在 30% 的数据，保证注释没有跨行情况。

对于 100% 的数据，保证代码总长度不超过 20KB，所有字符都是常用可见字符或者空格、tab、换行符，并且不含 ASCII 标准之外的字符（如中文、拓展 ASCII 表等等），所有测试数据中的程序均可用 C/C++ 编译器编译运行。保证以字符 `#` 作用的行（包含 `include`、宏定义相关语句）中，引号成对出现，并且不包含以 `\` 结尾的单行注释；单行注释若出现跨行情况，必定是 `\` 字符之后紧跟换行符。

请注意考虑各种字符串情况，诸如：

1. 字符串转义： `"/*\n//*/"`
2. 字符常量： `'//'''`

偏差 (deviation)

【题目背景】

wangyurzee7 和 Yazid 是好朋友。

【题目描述】

wangyurzee7 是一个繁忙的工作者，他每天都要处理许多有趣、复杂的问题。但由于能力有限，他总会出各种各样的偏差，这让 Yazid 头疼不已。

有一天，wangyurzee7 收到了一个任务：他获得了一个长度为 n 的序列 A （下标从 1 开始），他需要选择这个序列中的一个长度为 m 的连续子区间，然后把这段区间内的数按顺序写下来，得到序列 B 。

这个任务非常简单，但 wangyurzee7 实在是太弱了，所以他在做的时候还是出了偏差。他有一个偏差值 k ，当他在抄写得到序列 B 的时候，他把 B 中的每个元素都加上了 k （其中 k 是一个整数）！

也就是说，假设原序列为 $A[1] \dots A[n]$ ，wangyurzee7 取的子区间是 $[l, l+m-1]$ ，那么对于 $1 \leq i \leq m$ ，都有 $B[i] = A[l+i-1] + k$ 。

Yazid 勃然大怒，他根本没想到这么简单的任务 wangyurzee7 都不能胜任。他把 wangyurzee7 狠狠地批判了一番。

当然啦，责任还是要由出偏差的人来承担。可是 wangyurzee7 并不记得他的偏差值 k 。无奈之下，他只好退而求其次，提出了一些更模糊的问题：

1. 偏差值 k 的取值有几种可能。
2. 偏差值绝对值 $|k|$ 的最小值是多少。
3. 他选择的子区间的左端点 l 有几种取值可能。
4. 他选择的子区间的左端点 l 最左可能是多少。
5. 他选择的子区间的左端点 l 最右可能是多少。

既然要求的東西變少了，問題也就變得更簡單了。請你幫幫可憐的 wangyurzee7 解決這些問題。

【輸入格式】

從標準輸入讀入數據。

本題包含多組數據。第一行一個正整數 T 表示數據組數。接下來依次描述每組數據。對於每組數據：

第一行一個正整數 n ，表示序列 A 的長度。

第二行 n 個用空格隔開的非負整數 $A[1] \dots A[n]$ ，描述了序列 A 。

第三行一個正整數 m ，表示序列 B 的長度。

第四行 m 個用空格隔開的非負整數 $B[1] \dots B[m]$ ，描述了序列 B 。

【输出格式】

输出到标准输出。

对于每组数据，输出 5 个用空格隔开的整数，依次表示 5 个问题的答案。特别地，对于问题 2,4,5，如果无解，请输出 0 作为答案。

【样例 1 输入】

```
4
5
2 3 3 3 3
2
6 7
10
1 1 3 2 1 3 2 1 2 2
3
1 3 2
5
100 200 300 900 1000
2
800 900
4
2 3 3 3
2
1 233
```

【样例 1 输出】

```
1 4 1 1 1
1 0 2 2 5
3 100 3 1 4
0 0 0 0 0
```

【子任务】

对于 30% 的数据，保证 $n \leq 100$ ，序列中元素的值不超过 1,000。

对于 50% 的数据，保证 $n \leq 1,000$ 。

对于另外 20% 的数据，保证序列中元素的值不超过 100。

对于 100% 的数据，保证 $T \leq 6, 1 \leq m \leq n \leq 10^5$ ，序列中元素的值不超过 10^9 。