

随意组合

小明被绑架到 X 星球的巫师 W 那里。

其时，W 正在玩弄两组数据 (2 3 5 8) 和 (1 4 6 7)

他命令小明从一组数据中分别取数与另一组中的数配对，共配成 4 对（组中的每个数必被用到）。

小明的配法是：{(8,7),(5,6),(3,4),(2,1)}

巫师凝视片刻，突然说这个配法太棒了！

因为：

每个配对中的数字组成两位数，求平方和，无论正倒，居然相等：

$$87^2 + 56^2 + 34^2 + 21^2 = 12302$$

$$78^2 + 65^2 + 43^2 + 12^2 = 12302$$

小明想了想说：“这有什么奇怪呢，我们地球人都知道，随便配配也可以啊！”

{(8,6),(5,4),(3,1),(2,7)}

$$86^2 + 54^2 + 31^2 + 27^2 = 12002$$

$$68^2 + 45^2 + 13^2 + 72^2 = 12002$$

巫师顿时凌乱了……

请你计算一下，包括上边给出的两种配法，巫师的两组数据一共有多少种配对方案具有该特征。

配对方案计数时，不考虑配对的出现次序。

就是说：

{(8,7),(5,6),(3,4),(2,1)}

与

{(5,6),(8,7),(3,4),(2,1)}

是同一种方案。

注意：需要提交的是一个整数，不要填写任何多余内容（比如，解释说明文字等）

拼棋盘

有 8x8 和 6x6 的棋盘两块（棋盘厚度相同，单面有棋盘，背面无图案）。参见【图 1.png】组成棋盘的小格子是同样大小的正方形，黑白交错排列。

现在需要一个 10x10 的大棋盘，希望能通过锯开这两个棋盘，重新组合出大棋盘。

要求:

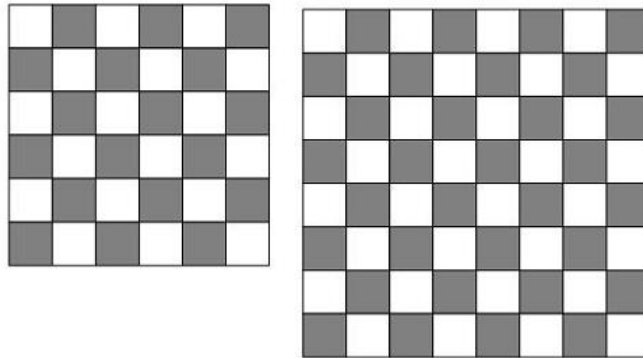
1. 拼好的大棋盘仍然保持黑白格交错的特性.
2. 两个已有的棋盘都只允许锯一锯 (即锯开为两块), 必须沿着小格的边沿, 可以折线锯开.
3. 要尽量保证 8x8 棋盘的完整, 也就是说, 从它上边锯下的那块面积要尽可能小.

要求提交的数据是: 4 块锯好的部分的面积, 按从小到大排列, 用空格分开.
(约定每个小格的面积为 1)

比如: 10 10 26 54

当然, 这个不是正确答案.

请严格按照要求格式提交数据, 不要填写任何多余的内容 (比如, 说明解释等)



打靶

小明参加 X 星球的打靶比赛,
比赛使用电子感应计分系统. 其中有一局, 小明得了 96 分.

这局小明共打了 6 发子弹, 没有脱靶.
但望远镜看过去, 只有 3 个弹孔.
显然, 有些子弹准确地穿过了前边的弹孔.

不同环数得分是这样设置的:

1,2,3,5,10,20,25,50

那么小明的 6 发子弹得分都是多少呢？有哪些可能情况呢？

下面的程序解决了这个问题。

仔细阅读分析代码，填写划线部分缺失的内容。

```
#include <stdio.h>
#define N 8

void f(int ta[], int da[], int k, int ho, int bu, int sc)
{
    int i,j;
    if(ho<0 || bu<0 || sc<0) return;
    if(k==N){
        if(ho>0 || bu>0 || sc>0) return;
        for(i=0; i<N; i++){
            for(j=0; j<da[i]; j++)
                printf("%d ", ta[i]);
        }
        printf("\n");
        return;
    }

    for(i=0; i<=bu; i++){
        da[k] = i;
        f(ta, da, k+1, _____, bu-i, sc-ta[k]*i); //填空位置
    }

    da[k] = 0;
}

int main()
{
    int ta[] = {1,2,3,5,10,20,25,50};
    int da[N];
    f(ta, da, 0, 3, 6, 96);
    return 0;
}
```

注意：只填写划线处缺少的内容，不要填写已有的代码或符号，也不要填写任何解释说明文字等。

路径之谜

小明冒充 X 星球的骑士，进入了一个奇怪的城堡。
城堡里边什么都没有，只有方形石头铺成的地面。

假设城堡地面是 $n \times n$ 个方格。【如图 1.png】所示。

按习俗，骑士要从西北角走到东南角。
可以横向或纵向移动，但不能斜着走，也不能跳跃。
每走到一个新方格，就要向正北方和正西方各射一箭。
(城堡的西墙和北墙内各有 n 个靶子)

同一个方格只允许经过一次，但不必走完所有的方格。

如果只给出靶子上箭的数目，你能推断出骑士的行走路线吗？

有时是可以的，比如图 1.png 中的例子。

本题的要求就是已知箭靶数字，求骑士的行走路径（测试数据保证路径唯一）

输入：

第一行一个整数 $N(0 < N < 20)$ ，表示地面有 $N \times N$ 个方格

第二行 N 个整数，空格分开，表示北边的箭靶上的数字（自西向东）

第三行 N 个整数，空格分开，表示西边的箭靶上的数字（自北向南）

输出：

一行若干个整数，表示骑士路径。

为了方便表示，我们约定每个小格子用一个数字代表，从西北角开始编号: 0,1,2,3....

比如，图 1.png 中的方块编号为：

```
0  1  2  3
4  5  6  7
8  9 10 11
12 13 14 15
```

示例：

用户输入：

4

2 4 3 4

4 3 3 3

程序应该输出:
0 4 5 1 2 3 7 11 10 9 13 14 15

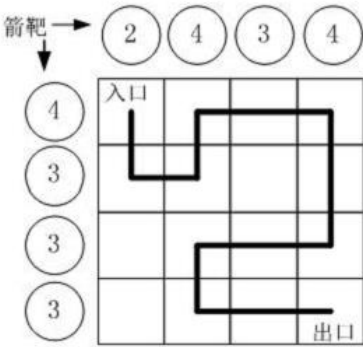
资源约定:
峰值内存消耗 < 256M
CPU 消耗 < 1000ms

请严格按照要求输出，不要画蛇添足地打印类似：“请您输入...” 的多余内容。

所有代码放在同一个源文件中，调试通过后，拷贝提交该源码。

注意: main 函数需要返回 0
注意: 只使用 ANSI C/ANSI C++ 标准，不要调用依赖于编译环境或操作系统的特殊函数。
注意: 所有依赖的函数必须明确地在源文件中 #include <xxx>， 不能通过工程设置而省略常用头文件。

提交时，注意选择所期望的编译器类型。



碱基

生物学家正在对 n 个物种进行研究。
其中第 i 个物种的 DNA 序列为 $s[i]$ ，其中的第 j 个碱基为 $s[i][j]$ 。碱基一定是 A、T、G、C 之一。
生物学家想找到这些生物中一部分生物的一些共性，他们现在关注那些至少在 m 个生物中

出现的长度为 k 的连续碱基序列。准确的说，科学家关心的序列用 $2m$ 元组 $(i_1, p_1, i_2, p_2, \dots, i_m, p_m)$ 表示，

满足：

$1 \leq i_1 < i_2 < \dots < i_m \leq n$;

且对于所有 $q (0 \leq q < k)$, $s[i_1][p_1+q] = s[i_2][p_2+q] = \dots = s[i_m][p_m+q]$ 。

现在给定所有生物的 DNA 序列，请告诉科学家有多少的 $2m$ 元组是需要关注的。如果两个 $2m$ 元组有任何一个位置不同，则认为不同的元组。

【输入格式】

输入的第一行包含三个整数 n 、 m 、 k ，两个整数之间用一个空格分隔，意义如题目所述。

接下来 n 行，每行一个字符串表示一种生物的 DNA 序列。

DNA 序列从 1 至 n 编号，每个序列中的碱基从 1 开始依次编号，不同的生物的 DNA 序列长度可能不同。

【输出格式】

输出一个整数，表示关注的元组个数。

答案可能很大，你需要输出答案除以 1000000007 的余数。

【样例输入】

3 2 2

ATC

TCG

ACG

【样例输出】

2

再例如：

【样例输入】

4 3 3

AAA

AAAA

AAA

AAA

【样例输出】

7

【数据规模与约定】

对于 20% 的数据， $k \leq 5$ ，所有字符串总长 L 满足 $L \leq 100$

对于 30% 的数据， $L \leq 10000$

对于 60% 的数据， $L \leq 30000$

对于 100% 的数据, $n \leq 5, m \leq 5, 1 \leq k \leq L \leq 100000$
保证所有 DNA 序列不为空且只会包含 'A' 'G' 'C' 'T' 四种字母

资源约定:
峰值内存消耗 < 256M
CPU 消耗 < 1000ms

请严格按照要求输出, 不要画蛇添足地打印类似: “请输入...” 的多余内容。

所有代码放在同一个源文件中, 调试通过后, 拷贝提交该源码。

注意: main 函数需要返回 0
注意: 只使用 ANSI C/ANSI C++ 标准, 不要调用依赖于编译环境或操作系统的特殊函数。
注意: 所有依赖的函数必须明确地在源文件中 #include <xxx>, 不能通过工程设置而省略常用头文件。

提交时, 注意选择所期望的编译器类型。

圆圈舞

春天温暖的阳光照耀着大地, 正是草原上的小动物们最快乐的时候。小动物们在草原上开了一个舞会, 欢度这美好的时光。

舞会上最重要的一个环节就是跳圆舞曲, n 只小动物手拉手围成一大圈, 随着音乐跳起来。在跳的过程中, 小动物们可能会变换队形。它们的变换方式是动物 A 松开自己右手, 动物 B 松开自己的左手, 动物 A 和 B 手拉到一起, 而它们对应的松开的手 (如果有的话) 也拉到一起。

例如, 假设有 10 只小动物, 按顺序围成一圈, 动物 1 的右手拉着动物 2 的左手, 动物 2 的右手拉着动物 3 的左手, 依次类推, 最后动物 10 的右手拉着动物 1 的左手。如果通过动物 2 和 8 变换队形, 则动物 2 的右手拉着动物 8 的左手, 而对应的动物 3 的左手拉着动物 7 的右手, 这样形成了 1-2-8-9-10 和 3-4-5-6-7 两个圈。如果此时通过动物 2 和 6 变换队形, 则将形成 1-2-6-7-3-4-5-8-9-10 一个大圈。注意, 如果此时通过动物 1 和 2 变换队形, 那么队形不会改变, 因为动物 1 的右手和动物 2 的左手松开后又拉到一起了。

在跳舞的过程中, 每个动物 i 都有一个欢乐值 H_i 和一个感动值 F_i 。
如果两个动物在一个圈中, 欢乐值会彼此影响, 产生欢乐能量。如果两个动物 i, j ($i \neq j$) 在同一个大小为 t 的圈中, 而动物 i 在动物 j 右手的第 p 个位置 (动物 j 右手的第 1 个位置就是动物 j 右手所拉着的动物, 而第 2 个位置就是右手第 1 个位置的动物右手拉着的动物, 依次类推), 则产生的欢乐能量为 $(t-p) \cdot H_j \cdot F_i$ 。在跳舞的过程中, 动物们的欢乐值和感动值有可能发生变化。

圆舞曲开始的时候, 所有的动物按编号顺序围成一个圈, 动物 n 右手的第 i 个位置正好是动

物 i 。现在已知小动物们变换队形的过程和欢乐值、感动值变化的过程，求每次变换后所有动物所产生的欢迎能量之和。

【输入格式】

输入的第一行包含一个整数 n ，表示动物的数量。

接下来 n 行，每行两个用空格分隔的整数 H_i, F_i ，按编号顺序给出每只动物的欢乐值和感动值。

接下来一行包含一个整数 m ，表示队形、欢乐值、感动值的变化次数。

接下来 m 行，每行三个用空格分隔的整数 k, p, q ，当 $k=1$ 时，表示小动物们通过动物 p 和动物 q 变换了队形，当 $k=2$ 时，表示动物 p 的欢乐值变为 q ，当 $k=3$ 时，表示动物 p 的感动值变为了 q 。

【输出格式】

输出 m 行，每行一个整数，表示每次变化后所有动物产生的能量之和。

答案可能很大，你需要计算答案除以 1000000007 的余数。

【样例输入】

```
10
1 1
1 1
1 1
1 1
1 1
1 1
1 1
1 1
1 1
1 1
9
1 2 8
1 2 6
2 8 10
3 5 10
1 1 2
1 2 1
2 5 5
1 4 8
1 4 5
```

【样例输出】

```
100
450
855
1341
```


1341
811
923
338
923

【数据规模与约定】

对于 20%的数据, $2 \leq n, m \leq 100$.

对于 30%的数据, $2 \leq n, m \leq 1000$.

另有 20%的数据, 只有 $k=1$ 的操作且 H_i, F_i 均为 1.

另有 20%的数据, 只有 $k=1$ 或 2 的操作且 F_i 均为 1.

对于 100%的数据, $2 \leq n, m \leq 100000$, $0 \leq H_i, F_i \leq 10^9$, $1 \leq k \leq 3$, $k=1$ 时 $1 \leq p, q \leq n$ 且 $p \neq q$, $k=2$ 或 3 时 $1 \leq p \leq n$ 且 $0 \leq q \leq 10^9$.

资源约定:

峰值内存消耗 < 256M

CPU 消耗 < 2500ms

请严格按照要求输出, 不要画蛇添足地打印类似: “请输入...” 的多余内容.

所有代码放在同一个源文件中, 调试通过后, 拷贝提交该源码.

注意: main 函数需要返回 0

注意: 只使用 ANSI C/ANSI C++ 标准, 不要调用依赖于编译环境或操作系统的特殊函数.

注意: 所有依赖的函数必须明确地在源文件中 `#include <xxx>`, 不能通过工程设置而省略常用头文件.

提交时, 注意选择所期望的编译器类型.