NOI Day 2

题目名称	弱	序列	黑心商人
题目类型	传统型	传统型	传统型
目录	`ro`	`seq`	`merchant.cpp`
可执行文件名	`ro`	`seq`	`merchant.cpp`
输入文件名	`ro.in`	`seq.in`	`merchant.in`
输出文件名	`ro.out`	`seq.out`	`merchant.out`
每个测试点时限	1.0 秒	5.0 秒	1.0 秒
内存限制	1024 Mib	1024 Mib	1024 Mib
测试点数目	20	20	7
测试点是否等分	是	是	否
提交源文件程序名	`ro.cpp`	`seq.cpp`	`merchant.cpp`

注意事项与提醒（请选手务必仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
3. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
4. 程序可使用的栈内存空间限制与题目的内存限制一致。

弱(ro)

题面描述

小弱是一只喜爱统计的青蛙。

这天，小弱在一个巨大的棋盘（你可以认为是无穷大）上移动一颗棋子。

具体来说，小弱在坐标 $(0,0)$ 处放了一颗棋子。然后，小弱进行了 n 次移动，每次以 $w_U : w_D : w_L : w_R$ 的概率往上下左右四个方向中的一个移动一单位。

现在，小弱想要知道，移动结束后棋子移动到不同坐标的数量的期望。

本来这个问题小弱可以轻松解答，但出于对统计学的热爱，它还想知道棋子移动到的不同坐标的数量的方差。

由于昨晚在赶 SCI 论文的 ddl，小弱现在非常困，所以它把这个问题交给了你。希望你能给出答案，并在乘上 $(w_U + w_D + w_L + w_R)^{2n}$ 后对 998244353 取模。

关于方差的解释：

$$\sigma^2=\sum_{i=1}^{n}P(X=X_i)(X_i-E(X))^2$$

输入格式

从文件 `ro.in` 中读入数据。

输入第一行一个整数 n，第二行四个整数 w_U,w_D,w_L,w_R。

输出格式

输出到文件 `ro.out` 中。

输出一行一个整数表示答案。

样例 1 输入

```
2
1 1 1 1
```

样例 1 输出

```
48
```

其他样例

参见附加文件。

数据范围与提示

对于所有数据：$1\le n\le 100$，$0\le w_U,w_D,w_L,w_R\le 100$，$1\le w_U+w_D+w_L+w_R$。

测试点编号	$n\le$	特殊性质
$1\sim 2$	10	
3	30	
4	50	
5	70	
$6\sim 7$	100	$w_L=w_R=0$
8	100	$w_U=w_D=w_L=w_R$
$9\sim 10$	100	

序列(seq)

问题描述

给出一个长度为 n 的序列 x 和定值 m，保证对于 $1 \le i \le n$，都有 $0 \le x_i \le m$。

考虑如下函数：

```
long long query(){
    vector<int>v(m+1);
    for(int i=1;i<=m;i++) v[i]=0;
    for(int i=1;i<=n;i++){
        for(int j=1;j<=x[i];j++) v[j]=max(0,v[j]-1);
        for(int j=x[i]+1;j<=m;j++) v[j]=v[j]+1;
    }
    long long ans=0;
    for(int i=1;i<=m;i++) ans+=v[i];
    return ans;
}
```

现在给出 q 个修改操作，第 i 个修改操作给出 u_i, v_i，表示将 x_{u_i} 修改为 v_i，每次修改完成后，你需要输出 $\text{query}()$ 的结果。

输入格式

第一行给出三个正整数 n, m, q。

第二行给出 n 个整数，表示 x_i，保证 $0 \le x_i \le m$。

之后 q 行，每行给出两个整数，表示 u_i, v_i，保证 $1 \le u_i \le n, 0 \le v_i \le m$。

输出格式

输出 q 行，对于每个询问，输出一个整数，表示答案。

样例输入1

```
3 4 2
1 2 3
1 4
3 0
```

样例输出1

```
2
6
```

样例解释1

第一次修改后，序列变成 $(4,2,3)$，调用函数后 $v=[0,0,0,2]$。

第二次修改后，序列变成 $(4,2,0)$，调用函数后 $v=[1,1,2,2]$。

样例输入2

```
7 2 9
2 0 2 2 0 1 0
1 1
3 0
4 0
4 1
6 1
3 2
2 0
3 2
2 0
```

样例输出2

```
4
7
11
9
9
6
6
6
6
```

样例输入3,4,5,6

见下发文件。

样例输出3,4,5,6

见下发文件。

评测数据规模

测试点编号	$n,m,q\le$
1	500
2	2000
$3,4,5,6$	10^5
$7,8,9,10$	5×10^5

黑心商人(merchant)

题目描述

高速公路上目前有 m 个收费站，编号为 1 到 m。在一天的不同时段，通过某个收费站的费用会有所不同。一天分为 n 个小时，编号为 1 到 n。目前，在第 i 个小时通过第 j 个收费站的花费为 $c_{i,j}$ 。其中一些成本可能为 0（此时收费站免费），甚至为负（此时司机通过收费站会获得 $-c_{i,j}$ ）。

整条高速公路很短，一小时就能走完。当然，你也不必如此匆忙，你可以在行驶过程中随意停车。不过，你不能在高速公路上过夜，必须在当天通过所有收费站。

当然，司机希望以尽可能低的花费通过高速公路。对于 $1 \le i \le j \le n$，我们用 $f(i,j)$ 表示司机在第 i 小时通过第一个收费站，并在第 j 小时通过最后一个收费站的情况下，通过整条高速公路的最小可能花费。所有 $f(i,j)$ 都是被两国政府的和平条约提前设置好的，作为高速公路管理者你不能更改他们。

但是，只要保证保留第一个和最后一个收费站，$f(i,j)$ 的值保持不变，并且设置的所有费用都是 1 的整数倍的情况下，你可以自由修改通过各个收费站的花费，甚至取消某些收费站。

为了最小化高速公路维护的费用，你希望取消尽可能多的收费站。确定为了满足条约内容，最少需要保留多少收费站。

收费计划重组项目将分为两个阶段。在第一阶段，即初步设计阶段，只需找到最佳的收费站数量即可。但在第二阶段，即项目实施阶段，你还需要提供一份完整的收费站价格表计划。

输入格式

第一行三个整数 n,m,q，分别表示一天中的小时数，收费站数和描述项目阶段的一位。$q=0$ 表示项目处于第一阶段（初步设计），$q=1$ 表示项目处于第二阶段（实施阶段）。

接下来 n 行描述目前的收费情况，第 i 行包含 m 个整数 $c_{i,1},c_{i,2},\ldots,c_{i,m}$ $(-10^6 \le c_{i,j} \le 10^6)$，意义如题目描述。

输出格式

第一行输出一个整数 k $(2 \le k \le m)$，表示最少需要保留多少收费站，才能满足没有 $f(i,j)$ 改变。如果 $q=0$，输出仅包含这一行一个整数。

如果 $q=1$，接下来 n 行输出满足题目条件的最优价格计划。第 i 行包含 k 个整数 $d_{i,1},d_{i,2},\ldots,d_{i,k}$ $(-10^{12} \le d_{i,j} \le 10^{12})$。$d_{i,j}$ 表示在第 i 小时通过第 j 个收费站的新花费。

可以从题目限制知道，总可以确定一个绝对值不超过 10^{12} 且花费均为整数的计划。

输入输出样例 #1

输入 #1

```
3 6 1
-1 0 4 0 -3 0
-4 1 5 2 -5 2
-5 2 3 0 -2 2
```

输出 #1

```
3
0 0 0
0 1 0
0 0 0
```

样例 1 解释

$f(i,j)$ 如下：
$f(1,1)=(-1)+0+4+0+(-3)+0=0$
$f(1,2)=(-1)+0+4+0+(-5)+2=0$
$f(1,3)=(-1)+0+4+0+(-5)+2=0$
$f(2,2)=(-4)+1+5+2+(-5)+2=1$
$f(2,3)=(-4)+1+3+0+(-2)+2=0$
$f(3,3)=(-5)+2+3+0+(-2)+2=0$

两个收费站无法实现相同的花费。请注意，第一个和最后一个收费站是不能取消的，尽管根据输出的 $d_{i,j}$ 费用，这两个收费站是不收费的。

输入输出样例 #2

输入 #2

```
5 7 0
0 0 0 8 0 0 0
0 7 6 5 9 7 0
0 0 0 5 9 6 0
9 4 0 4 4 7 0
0 0 0 9 8 6 0
```

输出 #2

```
3
```

样例 2 解释

在第二个样例中，由于收费计划重组草案仅处于初步阶段，因此输出不包含新价目表的计划。

样例 3 ~ 8

见下发文件，分别满足子任务 $2,2,3,3,4,4$ 的限制。

子任务

对于所有测试数据保证：

- $2 \le n,m \le 3\times 10^4$;
- $n\times m \le 3\times 10^5$;
- $q\in\{0,1\}$;

- $|c_{i,j}| \le 10^6$。

本题采用捆绑测试。 共包含 7 个子任务：

- 子任务 1（12 分）：保证 $n, m \le 7$，$|c_{i,j}| \le 15$，$q = 0$；
- 子任务 2（16 分）：保证 $n \times m \le 60^2$，$q = 0$；
- 子任务 3（16 分）：保证 $n, m \le 400$，$q = 0$；
- 子任务 4（14 分）：保证 $q = 0$；
- 子任务 5（12 分）：保证 $n, m \le 7$，$|c_{i,j}| \le 15$；
- 子任务 6（16 分）：保证 $n \times m \le 60^2$；
- 子任务 7（16 分）：无特殊限制。