题目名称	ckr 与平方数	单位根	病毒
题目类型	传统型	传统型	传统型
目录	sqr	math	viru
可执行文件名	sqr	math	viru
输入文件名	sqr.in	math.in	viru.in
输出文件名	sqr.out	math.out	viru.out
每个测试点时限	1 秒	1 秒	4 秒
内存限制	512 MiB	1024 MiB	512 MiB
子任务数目	10	25	9
测试点是否等分	是	是	否

提交源程序文件名

对于 C++ 语言	sqr.cpp	math.cpp	viru.cpp

编译选项

对于 C++ 语言	-O2 -std=c++14

ckr 与平方数（sqr）

【题目描述】

众所周知，ckr 的微积分水平很不怎么样，就像现在他就不会求一个很水的积分：$\int (x+1)^n \,\mathrm{d}x$ 的封闭形式而只会暴力展开一项项积分。

不过同样众所周知，ckr 很擅长枚举和找规律，于是他发现 $\int (x+1)^n \,\mathrm{d}x = \dfrac{(x+1)^{n+1}}{n+1} + C$。

更进一步，他发现 $\int (x+t)^n \,\mathrm{d}x = \dfrac{(x+t)^{n+1}}{n+1} + C$，解决了这一整类的积分。

于是他想要寻找更一般的规律，开始研究 $\int (x+s)^n (x+t)^m \,\mathrm{d}x$，由于 ckr 很喜欢平方数，所以他决定先研究 n, m 是平方数的情况。

当然，由于不定积分比较麻烦，所以你只需要输出定积分：$\int_0^{x_0} (x+s)^n (x+t)^m \,\mathrm{d}x$，同时 n, m 也不可能太大，所以 ckr 要求 $1 \le n, m \le N$。

由于 ckr 现在还处于探索阶段，每一个数据都是很重要的，所以你需要对每个 $1 \le n, m \le N$ 且 n, m 是完全平方数求出答案。

由于答案可能很大，你只需要求出答案关于 2147483647 取模后的值。

【输入格式】

从文件 ***sqr.in*** 中读入数据。

输入一行四个整数 N, s, t, x_0。

【输出格式】

输出到文件 ***sqr.out*** 中。

输出 $\lfloor \sqrt{N} \rfloor$ 行，每行 $\lfloor \sqrt{N} \rfloor$ 个整数，其中第 i 行第 j 个整数表示 $n = i^2, m = j^2$ 的答案。

【样例 1 输入】

```
4 3 7 1
```

【样例 1 输出】

```
1431655791 1932746596
930577433 1950302041
```

【样例 1 解释】

分数形式的答案为：$\begin{bmatrix} \dfrac{79}{3} & \dfrac{113137}{10} \\ \dfrac{35579}{30} & \dfrac{164819873}{315} \end{bmatrix}$。

【样例 2 输入】

```
9 432626436 222345443 0
```

【样例 2 输出】

```
0 0 0
0 0 0
0 0 0
```

【样例 3 输入】

```
9 233333 233333 666666
```

【样例 3 输出】

```
1703229151 2113117123 1350295746
2113117123 1384164355 1123817829
1350295746 1123817829 456733368
```

【数据范围】

对于 10% 的测试点，保证 $1 \le N \le 10$。
对于 30% 的测试点，保证 $1 \le N \le 100$。
对于 60% 的测试点，保证 $1 \le N \le 3000$。
另有 10% 的测试点，保证 $s = t$。
对于 100% 的测试点，保证 $0 \le s, t, x_0 < 2147483647, 1 \le N \le 10^5$。

【Ckr 教你学数学】

你可能会用到以下公式：

1. 多项式定积分的计算方法

   对多项式 $f(x) = \sum_{i=0}^{n} c_i x^i$，定积分 $\int_a^b f(x)\mathrm{d}x$ 可以用如下公式计算：

   $$\int_a^b f(x)\,\mathrm{d}x = \sum_{i=0}^{n} \frac{(b^{i+1} - a^{i+1})c_i}{i+1}$$

2. 换元积分法

设 $I \subseteq \mathbb{R}$ 为一个区间，$\varphi : [a,b] \to I$ 是一个导数可积的函数。设 $f : I \to \mathbb{R}$ 是一个连续函数，则：

$$\int_{\varphi(a)}^{\varphi(b)} f(u)\,\mathrm{d}u = \int_a^b f(\varphi(x))\varphi'(x)\,\mathrm{d}x.$$

3. 分部积分法

设 $u = u(x), \mathrm{d}u = u'(x)\mathrm{d}x$，与 $v = v(x), \mathrm{d}v = v'(x)\mathrm{d}x$，则：

$$\int_a^b u(x)v'(x)\,\mathrm{d}x = [u(x)v(x)]_a^b - \int_a^b u'(x)v(x)\,\mathrm{d}x = u(b)v(b) - u(a)v(a) - \int_a^b u'(x)v(x)\,\mathrm{d}x$$

【温馨提示】

在本题中，如果你希望获得全部的分数，你可能考虑由于常数的影响。只使用加法、减法和乘法运算不会有太大的常数，但请谨慎使用除法。

1. 跨越比较大的 2 的幂的数组寻址会产生较大的常数。

2. 过多的分支可以带来相当大的常数。

当然，如果你的算法在数学上是正确的，但没有考虑常数的影响，可能仍然可以获得一部分的分数。

单位根（math）

【题目描述】

S 省 Θ 中学的高 e (i) 班正在上数学课。

在数学课上，老师介绍了复数和单位根，并介绍了单位根的一个性质：

多项式 x^n-1 在复数域 $\mathbb{C}$ 上共有 n 个根，它们可以表示成 $\omega_n^k=\cos\dfrac{2\pi k}{n}+\sin\dfrac{2\pi k}{n}\mathrm{i},k=0,1,\cdots,n-1$，其中 $\omega_n=\cos\dfrac{2\pi}{n}+\sin\dfrac{2\pi}{n}\mathrm{i}$。

对于正整数 n 和整数 k，这些单位根满足：

$$1+\omega_n^k+\omega_n^{2k}+\cdots+\omega_n^{(n-1)k}=\begin{cases}0 & n\nmid k\\ n & n\mid k\end{cases}$$

那天晚上的数学作业中，fy 遇到了这样一道题目：

> 给定 K 次多项式 $f(x)=a_0+a_1x+a_2x^2+\cdots+a_Kx^K$，设 $x^n=1$ 的 n 个根为 $r_1,r_2,\cdots,r_n$，求表达式 $\sum\limits_{i=1}^{n}f(r_i)$ 的值。

scx 看了一眼，就秒掉了这个问题：

> 将 $f(x)$ 的表达式代入，有
>
> $$\sum_{i=1}^{n}f(r_i)=\sum_{i=0}^{n-1}f\left(\omega_n^i\right)=\sum_{i=0}^{n-1}\sum_{j=0}^{K}a_j\omega_n^{ij}=\sum_{j=0}^{K}a_j\sum_{i=0}^{n-1}\omega_n^{ij}=\sum_{j=0}^{K}a_j\cdot\begin{cases}0 & n\nmid j\\ n & n\mid j\end{cases}=n\cdot\sum_{i=0}^{\lfloor K/n\rfloor}a_{i\cdot n}$$

scx 想，只有 n 个根，显得有些单调。于是，她把所有**不超过** n **次**的单位根都列在了一起，设为 $r_1,r_2,\cdots,r_m$。

她想求表达式 $\sum\limits_{i=1}^{m}f(r_i)$ 的值。

scx 注意到，对于一个复数，它可能会是多个 n 的 n 次单位根——比如 i 既是 4 次单位根，又是 8 次单位根。而重复计算同一个函数值，正是她不想要的。

于是，她将这些根去重后，写在了一起，设为 $r_1',r_2',\cdots,r_k'$，需要求表达式 $\sum\limits_{i=1}^{k}f(r_i')$ 的值。

然而根太多了，她并不能做出这道题。因此，她带着这个问题向班上的数学老师询问，不料数学老师外出有事，于是她带着草稿纸找到了在机房中的 fy，想让她帮忙算一下这个表达式的值。

【输入格式】

从文件 ***math.in*** 中读入数据。

第一行包含两个非负整数 n,K，表示单位根的最高次数和多项式的次数。

第二行包含 $K+1$ 个整数 $a_0,a_1,\cdots,a_K$，依次表示这些多项式的系数。

【输出格式】

输出到文件 *math.out* 中。

输出一行一个整数，表示表达式的值。可以证明，答案一定是一个整数，且在给定数据规模下，答案在带符号 128 位整数范围内。

【样例 1 输入】

```
4 1
-1 2
```

【样例 1 输出】

```
-8
```

【样例 1 解释】

不超过 4 次的互不相同的单位根有 6 个，分别是 $1, -1, \mathrm{i}, -\mathrm{i}, \dfrac{-1+\sqrt{3}\mathrm{i}}{2}, \dfrac{-1-\sqrt{3}\mathrm{i}}{2}$。

将它们代入 $f(x) = 2x - 1$，得

$$f(1)+f(-1)+f(\mathrm{i})+f(-\mathrm{i})+f\left(\frac{-1+\sqrt{3}\mathrm{i}}{2}\right)+f\left(\frac{-1-\sqrt{3}\mathrm{i}}{2}\right) = 1+(-3)+(-1+2\mathrm{i})+(-1-2\mathrm{i})+\left(-2+\sqrt{3}\mathrm{i}\right)+\left(-2-\sqrt{3}\mathrm{i}\right.$$

【样例 2 输入】

```
6 7
2 0 0 3 0 7 3 1
```

【样例 2 输出】

```
57
```

【样例 3】

见选手目录下的 *math/math3.in* 与 *math/math3.ans*。

【数据范围】

对于所有的测试点，均满足 $1 \le n \le 2^{31} - 1; 0 \le K \le 10^7; -9999 \le a_i \le 9999$。

<table>
<tr><th>测试点编号</th><th>n</th><th>K</th><th>其它性质</th></tr>
<tr><td>1</td><td>$=2$</td><td rowspan="4">$\le 10^5$</td><td rowspan="6">无</td></tr>
<tr><td>2</td><td>$=4$</td></tr>
<tr><td>3</td><td>$=6$</td></tr>
<tr><td>4</td><td>≤ 24</td></tr>
<tr><td>5</td><td>≤ 250</td><td rowspan="2">≤ 1000</td></tr>
<tr><td>6</td><td>≤ 1000</td></tr>
<tr><td>7</td><td>$\le 10^5$</td><td rowspan="6">$=0$</td><td rowspan="6">$f(x)=1$</td></tr>
<tr><td>8</td><td>$\le 10^6$</td></tr>
<tr><td>9</td><td>$\le 10^7$</td></tr>
<tr><td>10</td><td>$\le 10^8$</td></tr>
<tr><td>11</td><td>$\le 10^9$</td></tr>
<tr><td>12</td><td>$\le 2^{31}-1$</td></tr>
<tr><td>13</td><td>≤ 1000</td><td rowspan="5">$\le 10^5$</td><td>无</td></tr>
<tr><td>14</td><td rowspan="2">$\le 10^5$</td><td>$a_0=0$</td></tr>
<tr><td>15</td><td>无</td></tr>
<tr><td>16</td><td rowspan="2">$\le 5\times 10^6$</td><td>$a_0=0$</td></tr>
<tr><td>17</td><td>无</td></tr>
<tr><td>18</td><td rowspan="3">$\le 10^8$</td><td>$\le 10^7$</td><td>$f(x)=x^K$</td></tr>
<tr><td>19</td><td rowspan="2">$\le 10^5$</td><td>$a_0=0$</td></tr>
<tr><td>20</td><td>无</td></tr>
<tr><td>21</td><td rowspan="5">$\le 2^{31}-1$</td><td>$\le 10^7$</td><td>$f(x)=x^K$</td></tr>
<tr><td>22</td><td rowspan="2">$\le 10^5$</td><td>$a_0=0$</td></tr>
<tr><td>23</td><td rowspan="3">无</td></tr>
<tr><td>24</td><td rowspan="2">$\le 10^7$</td></tr>
<tr><td>25</td></tr>
</table>

病毒（viru）

【题目背景】

经过了一次病毒的袭击，ω 国拥有了丰富的经验。

而经过大发展的 ω 国的城市布局重新改变了，变为了 $n \times n$ 的一个大网格。

而现在，出现了新的紧急状况，经过小 ω 的调查，居然每个城市内都有病毒，由于其传播能力不强，所以没有大范围传播而被发现，第 i 行第 j 列的城市中含有病毒 $a_{i,j}$。

小 ω 决定彻底在 ω 国内消灭所有病毒，作为小 ω 的顶级秘书，你自然要帮助小 ω 做一些工作。

具体的，小 ω 将会先随机一个矩形（随机方式为在所有左上角是城市，右下角也是城市的矩形中等概率随机选取一个，边长可以是 0，边平行于网格的边），求出里面的病毒种数，然后让医生制造疫苗，你，要帮助小 ω 计算这个病毒的期望种数。

由于一些奇怪原因，你只要输出这个期望种数乘上 $\frac{n\times(n+1)\times n\times(n+1)}{4}$ 的值就好。

【输入格式】

从文件 *viru.in* 中读入数据。

第一行一个正整数 n 表示网格的范围。

下面 n 行，每行 n 个正整数 $a_{i,j}$ 表示坐标位于 (i,j) 的城市中所含有的病毒。

【输出格式】

输出到文件 *viru.out* 中。

一行一个数表示期望种数乘上 $\frac{n\times(n+1)\times n\times(n+1)}{4}$ 的值，由于得便的力量，它总是一个整数。

【样例 1 输入】

```
5
2 4 2 2 4
5 5 3 2 3
3 3 4 2 1
4 2 3 2 3
3 3 5 5 2
```

【样例 1 输出】

```
644
```

【样例 2 输入】

```
15
9 9 1 1 6 4 4 5 8 6 10 6 1 7 9
5 8 10 3 9 1 8 6 10 7 1 4 2 2 4
9 6 1 6 10 9 7 5 3 9 9 6 7 10 1
5 7 6 5 2 8 8 7 2 10 2 5 3 7 5
10 7 10 7 6 10 7 2 5 6 4 10 5 6 6
6 2 5 10 7 2 6 8 5 6 2 10 9 3 2
3 9 1 5 2 1 7 9 7 10 6 7 2 5 7
3 9 2 10 5 1 3 8 8 3 6 4 5 5 6
7 3 8 9 3 8 10 7 3 6 5 1 10 2 6
9 10 6 5 5 4 10 7 10 6 9 4 9 5 8
8 9 7 2 6 6 2 6 6 3 5 3 10 6 1
10 9 10 9 7 10 5 8 3 9 1 8 9 10 3
4 9 2 1 8 10 2 9 4 5 4 5 10 3 8
4 5 7 3 5 6 6 3 2 6 7 4 1 3 3
1 5 2 9 2 7 2 1 3 10 10 5 7 9 9
```

【样例 2 输出】

```
112292
```

【数据范围】

对于所有数据，保证 $1 \leq n \leq 1500$。

数据全部随机生成：随机方式为手动选取一个正整数 W 满足 $1 \leq W \leq n \times n$，每个 $a_{i,j}$ 都在 $[1, W]$ 中随机生成。

【子任务】

Subtask 1(23): $1 \leq n \leq 50$。
Subtask 2(19): $1 \leq n \leq 100$。
Subtask 3(17): $1 \leq n \leq 150$。
Subtask 4(13): $1 \leq n \leq 300$。
Subtask 5(11): $1 \leq n \leq 600$。
Subtask 6(7): $1 \leq n \leq 800$。
Subtask 7(5): $1 \leq n \leq 1000$。
Subtask 8(3): $1 \leq n \leq 1300$。
Subtask 9(2): 无。