Crazy Bacteria

Time limit per test: 0.5 seconds
Memory limit per test: 1 gb

Doctor B is experimenting on a newly discovered Bacteria known as “Crazy Bacteria”. The name comes from the fact that under optimal conditions, this bacteria reproduces and moves around at a tremendous rate. Doctor B believes that understanding how this bacteria becomes super charged by just a change in conditions could allow humans to make a lot of breakthroughs. As such, he wants to keep track of the bacteria while providing them these ideal conditions.

Doctor B is a little unconventional and uses rectangle shaped petri dishes to do his experiments. He places the bacteria in the petri dishes as evenly as possible and starts providing them the conditions they require. However, he quickly realizes that he’s not able to keep track of them at all. Therefore, he asks you for help.

The petri dishes can be seen as an N x M grid where each cell holds $C_{i,j}$ (1 <= i <= N, 1 <= j <= M) bacteria. In one iteration, the bacteria in a single cell do the following:

1) Move x cells rightwards if x is odd and x cells leftwards if x is even (where x is the iteration number starting from 1). While going rightwards, if a row ends, then the bacteria go to the first column of the next row and if it’s the last row, the bacteria go to the first column of the first row. While going leftwards, if a row ends, then the bacteria go to the last column of the previous row and if it’s the first row, the bacteria go to the last column of the last row. For example see the following example showing only the first step in each iteration:

<table>
<tr><td>Before iteration 1’s first step:</td><td>After iteration 1’s first step:</td></tr>
<tr><td>
<table>
<tr><td>3</td><td>1</td><td>5</td></tr>
<tr><td>7</td><td>9</td><td>12</td></tr>
<tr><td>14</td><td>55</td><td>2</td></tr>
</table>
</td><td>
<table>
<tr><td>2</td><td>3</td><td>1</td></tr>
<tr><td>5</td><td>7</td><td>9</td></tr>
<tr><td>12</td><td>14</td><td>55</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Before iteration 2’s first step:</th><th>After iteration 2’s first step:</th></tr>
<tr><td><table><tr><td>2</td><td>3</td><td>1</td></tr><tr><td>5</td><td>7</td><td>9</td></tr><tr><td>12</td><td>14</td><td>55</td></tr></table></td><td><table><tr><td>1</td><td>5</td><td>7</td></tr><tr><td>9</td><td>12</td><td>14</td></tr><tr><td>55</td><td>2</td><td>3</td></tr></table></td></tr>
</table>

<table>
<tr><th>Before iteration 3’s first step:</th><th>After iteration 3’s first step:</th></tr>
<tr><td><table><tr><td>1</td><td>5</td><td>7</td></tr><tr><td>9</td><td>12</td><td>14</td></tr><tr><td>55</td><td>2</td><td>3</td></tr></table></td><td><table><tr><td>55</td><td>2</td><td>3</td></tr><tr><td>1</td><td>5</td><td>7</td></tr><tr><td>9</td><td>12</td><td>14</td></tr></table></td></tr>
</table>

2) Explode! This causes the bacteria to spread copies of themselves in every cell except the current one. For example, a cell with 5 bacteria would cause a +5 (spread copies) in every other cell and a -5 (explode themselves) in the current cell. See the following example:

<table>
<tr><th>Before explosion:</th><th>After explosion:</th></tr>
<tr><td><table><tr><td>14</td><td>55</td><td>2</td></tr><tr><td>3</td><td>1</td><td>5</td></tr><tr><td>7</td><td>9</td><td>12</td></tr></table></td><td><table><tr><td>55 + 2 + 3 + 1 + 5 + 7 + 9 + 12</td><td>14 + 2 + 3 + 1 + 5 + 7 + 9 + 12</td><td>14 + 55 + 3 + 1 + 5 + 7 + 9 + 12</td></tr><tr><td>14 + 55 + 2 + 1 + 5 + 7 + 9 + 12</td><td>14 + 55 + 2 + 3 + 5 + 7 + 9 + 12</td><td>14 + 55 + 2 + 3 + 1 + 7 + 9 + 12</td></tr><tr><td>14 + 55 + 2 + 3 + 1 + 5 + 9 + 12</td><td>14 + 55 + 2 + 3 + 1 + 5 + 7 + 12</td><td>14 + 55 + 2 + 3 + 1 + 5 + 7 + 9</td></tr></table></td></tr>
</table>

Given an initial state of the grid and a number of iterations, your task is to print the final state of the grid after the given number of iterations.

Since the count of bacteria may quickly become very large, output it modulo 998244353.

Input:

The first line of input contains a single integer $\boldsymbol{T}(1<=\boldsymbol{T}<=1\text{x}10^{4})$ representing the number of test cases. This is followed by a description of each test case.

The first line of each test case contains three integers $\boldsymbol{N}(1<=\boldsymbol{N}<=1\text{x}10^{5})$, $\boldsymbol{M}$ $(1<=\boldsymbol{M}<=1\text{x}10^{5})$, and $\boldsymbol{I}$ $(0<=\boldsymbol{I}<=1\text{x}10^{18})$ representing the number of rows, number of columns, and the number of iterations to run, respectively.

The next $\boldsymbol{N}$ lines each contain $\boldsymbol{M}$ integers $\boldsymbol{C}_{i,j}(0<=\boldsymbol{C}_{i,j}<=1\text{x}10^{18}$, $1<=\boldsymbol{i}<=\boldsymbol{N}$, $1<=\boldsymbol{j}<=\boldsymbol{M})$ where $\boldsymbol{C}_{i,j}$ represents the number of bacteria in the $\boldsymbol{i}^{th}$ row and $\boldsymbol{j}^{th}$ column.

It is guaranteed that the sum of *NxM* across all cases will not exceed $1\text{x}10^{6}$.

Output:

Output $\boldsymbol{N}$ lines each containing $\boldsymbol{M}$ integers $\boldsymbol{D}_{i,j}$ $(0<=\boldsymbol{D}_{i,j}<=998244352$, $1<=\boldsymbol{i}<=\boldsymbol{N}$, $1<=\boldsymbol{j}<=\boldsymbol{M})$ where $\boldsymbol{D}_{i,j}$ represents the number of bacteria after $\boldsymbol{I}$ iterations in the $\boldsymbol{i}^{th}$ row and $\boldsymbol{j}^{th}$ column and modulo 998244353.

Example:

Input
1 3 3 3 14 55 2 3 1 5 7 9 12
Output

```
6147 6144 6142
6101 6154 6153
6155 6151 6149
```

Input
1 4 3 1918213 38492234598340 9234823234345 238475348957 234234231234985 14231413434534 32457893598 89838475934 563845738 2345789245 3457893458 34583409583 1234328798
Output
619164276 138510061 569392840 783938039 670078179 490136997 897154133 202647565 70269483 240045635 946457631 691300412