A - Lucky Direction

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 100 points

Problem Statement

You are given a string D representing one of the eight directions (north, east, west, south, northeast, northwest, southeast, southwest). The correspondence between the directions and their representing strings is as follows.

- North: `N`
- East: `E`
- West: `W`
- South: `S`
- Northeast: `NE`
- Northwest: `NW`
- Southeast: `SE`
- Southwest: `SW`

Print the string representing the direction opposite to the direction denoted by D.

Constraints

- D is one of `N`, `E`, `W`, `S`, `NE`, `NW`, `SE`, `SW`.

Input

The input is given from Standard Input in the following format:

```
D
```

Output

Print the answer.

Sample Input 1

```
N
```

Sample Output 1

```
S
```

Print `S`, which represents south, the direction opposite to north.

Sample Input 2

```
SE
```

Sample Output 2

```
NW
```

Print `NW`, which represents northwest, the direction opposite to southeast.

B - Seek Grid

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 200 points

Problem Statement

You are given an $N \times N$ grid S and an $M \times M$ grid T. The cell at the i-th row from the top and the j-th column from the left is denoted by (i, j).

The colors of the cells in S and T are represented by N^2 characters $S_{i,j}$ $(1 \leq i, j \leq N)$ and M^2 characters $T_{i,j}$ $(1 \leq i, j \leq M)$, respectively. In grid S, cell (i, j) is white if $S_{i,j}$ is `.`, and black if $S_{i,j}$ is `#`. The same applies for grid T.

Find T within S. More precisely, output integers a and b $(1 \leq a, b \leq N - M + 1)$ that satisfy the following condition:

- $S_{a+i-1,b+j-1} = T_{i,j}$ for every i, j $(1 \leq i, j \leq M)$.

Constraints

- $1 \leq M \leq N \leq 50$
- N and M are integers.
- Each of $S_{i,j}$ and $T_{i,j}$ is `.` or `#`.
- There is exactly one pair (a, b) satisfying the condition.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad M \\
S_{1,1}S_{1,2}\ldots S_{1,N} \\
S_{2,1}S_{2,2}\ldots S_{2,N} \\
\vdots \\
S_{N,1}S_{N,2}\ldots S_{N,N} \\
T_{1,1}T_{1,2}\ldots T_{1,M} \\
T_{2,1}T_{2,2}\ldots T_{2,M} \\
\vdots \\
T_{M,1}T_{M,2}\ldots T_{M,M}
\end{array}
$$

Output

Print a and b in this order, separated by a space on one line.

Sample Input 1

```
3 2
#.#
..#
##.
.#
#.
```

Sample Output 1

```
2 2
```

The 2×2 subgrid of S from the 2nd to the 3rd row and from the 2nd to the 3rd column matches T.

Sample Input 2

```
2 1
#.
##
.
```

Sample Output 2

```
1 2
```

C - Pigeonhole Query

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 300 points

Problem Statement

There are N pigeons numbered from 1 to N, and there are N nests numbered from 1 to N. Initially, pigeon i is in nest i for $1 \leq i \leq N$.

You are given Q queries, which you must process in order. There are two types of queries, each given in one of the following formats:

- `1 P H` : Move pigeon P to nest H.
- `2` : Output the number of nests that contain more than one pigeon.

Constraints

- $2 \leq N \leq 10^6$
- $1 \leq Q \leq 3 \times 10^5$
- $1 \leq P, H \leq N$
- For a query of the first type, pigeon P is not in nest H before the move.
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{aligned}
&N \quad Q \\
&\text{query}_1 \\
&\text{query}_2 \\
&\vdots \\
&\text{query}_Q
\end{aligned}
$$

Each query is given in one of the following two formats:

$$1 \quad P \quad H$$

$$2$$

Output

Print the answer to each query on a new line according to the instructions in the problem statement.

Sample Input 1

```
4 7
2
1 1 2
2
1 3 2
2
1 3 4
2
```

Sample Output 1

```
0
1
1
2
```

Initially, pigeons $1, 2, 3, 4$ are in nests $1, 2, 3, 4$, respectively.

- For the 1st query, the counts of pigeons in nests $1, 2, 3, 4$ are $1, 1, 1, 1$. No nests contain multiple pigeons, output 0.
- For the 2nd query, move pigeon 1 to nest 2.
- For the 3rd query, the counts become $0, 2, 1, 1$, respectively. One nest (nest 2) contains multiple pigeons, so output 1.
- For the 4th query, move pigeon 3 to nest 2.
- For the 5th query, the counts become $0, 3, 0, 1$, respectively. One nest (nest 2) contains multiple pigeons, so output 1.
- For the 6th query, move pigeon 3 to nest 4.
- For the 7th query, the counts become $0, 2, 0, 2$, respectively. Two nests (nests 2 and 4) contain multiple pigeons, so output 2.

Sample Input 2

```
5 10
2
1 4 3
1 4 5
2
1 3 1
2
1 2 3
1 2 5
1 1 3
2
```

Sample Output 2

```
0
1
2
1
```

D - Gravity

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 400 points

Problem Statement

There is a grid with 10^9 rows and W columns. The cell at the x-th column from the left and the y-th row from the **bottom** is denoted by (x, y).

There are N blocks. Each block is a 1×1 square, and block i-th ($1 \leq i \leq N$) is located at cell (X_i, Y_i) at time 0.

At times $t = 1, 2, \dots, 10^{100}$, the blocks are moved according to the following rules:

- If the entire bottom row is filled with blocks, then all blocks in the bottom row are removed.
- For each remaining block, in order from bottom to top, perform the following:
  - If the block is in the bottom row, or if there is a block in the cell immediately below it, do nothing.
  - Otherwise, move the block one cell downward.

You are given Q queries. For the j-th query ($1 \leq j \leq Q$), answer whether block A_j exists at time $T_j + 0.5$.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq W \leq N$
- $1 \leq X_i \leq W$
- $1 \leq Y_i \leq 10^9$
- $(X_i, Y_i) \neq (X_j, Y_j)$ if $i \neq j$.
- $1 \leq Q \leq 2 \times 10^5$
- $1 \leq T_j \leq 10^9$
- $1 \leq A_j \leq N$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \quad W \\
X_1 \quad Y_1 \\
X_2 \quad Y_2 \\
\vdots \\
X_N \quad Y_N \\
Q \\
T_1 \quad A_1 \\
T_2 \quad A_2 \\
\vdots \\
T_Q \quad A_Q
\end{array}
$$

Output

Print Q lines. The i-th line should contain `Yes` if block A_i exists at time $T_i + 0.5$, and `No` otherwise.

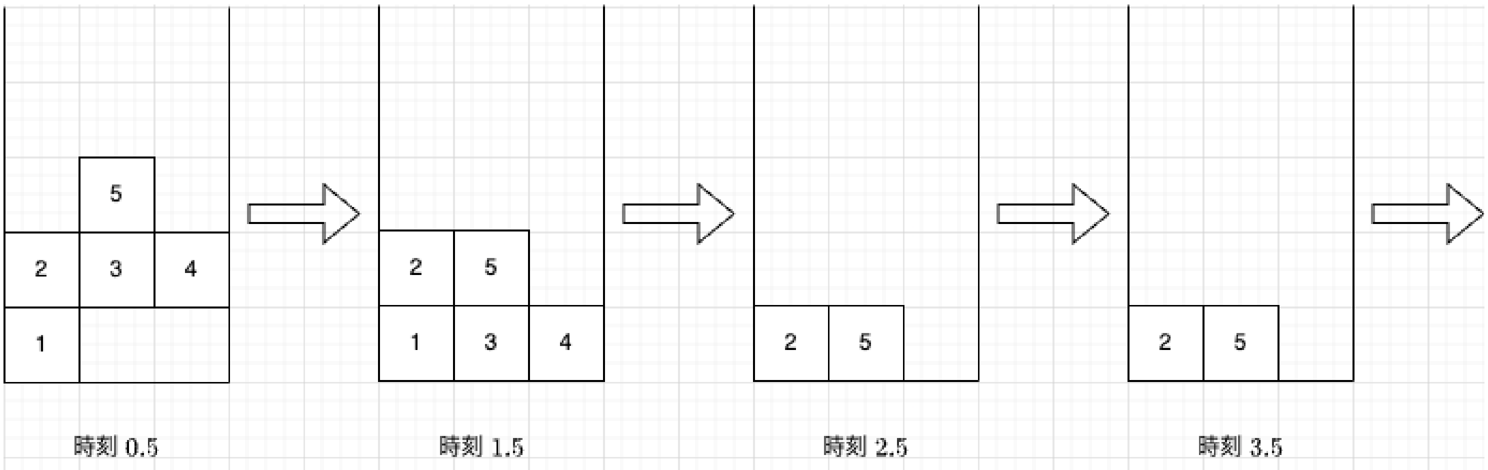
Sample Input 1

```
5 3
1 1
1 2
2 2
3 2
2 3
6
1 1
1 2
2 3
2 5
3 4
3 5
```

Sample Output 1

```
Yes
Yes
No
Yes
No
Yes
```

The positions of the blocks change as follows: ("時刻" means "time.")



- Query 1: At time 1.5, block 1 exists, so the answer is `Yes`.
- Query 2: At time 1.5, block 2 exists, so the answer is `Yes`.
- Query 3: Block 3 disappears at time 2, so it does not exist at time 2.5, and the answer is `No`.

Sample Input 2

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

Sample Output 2

```
No
No
Yes
Yes
```

E - Hierarchical Majority Vote

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 450 points

Problem Statement

For a binary string $B = B_1B_2\dots B_{3^n}$ of length 3^n ($n \geq 1$), we define an operation to obtain a binary string $C = C_1C_2\dots C_{3^{n-1}}$ of length 3^{n-1} as follows:

- Partition the elements of B into groups of 3 and take the majority value from each group. That is, for $i = 1, 2, \dots, 3^{n-1}$, let C_i be the value that appears most frequently among B_{3i-2}, B_{3i-1}, and B_{3i}.

You are given a binary string $A = A_1A_2\dots A_{3^N}$ of length 3^N. Let $A' = A'_1$ be the length-1 string obtained by applying the above operation N times to A.

Determine the minimum number of elements of A that must be changed (from 0 to 1 or from 1 to 0) in order to change the value of A'_1.

Constraints

- N is an integer with $1 \leq N \leq 13$.
- A is a string of length 3^N consisting of 0 and 1.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l}
N \\
A_1A_2\dots A_{3^N}
\end{array}
$$

Output

Print the answer.

Sample Input 1

```
2
010011101
```

Sample Output 1

```
1
```

For example, with $A = 010011101$, after applying the operation twice, we obtain:

- First operation: The majority of 010 is 0, of 011 is 1, and of 101 is 1, resulting in 011.
- Second operation: The majority of 011 is 1, yielding 1.

To change the final value from 1 to 0, one way is to change the 5th character of A from 1 to 0, yielding $A = 010001101$. After the change, the operations yield:

- First operation: The majority of 010 is 0, of 001 is 0, and of 101 is 1, resulting in 001.
- Second operation: The majority of 001 is 0, yielding 0.

Thus, the minimum number of changes required is 1.

Sample Input 2

```
1
000
```

Sample Output 2

```
2
```

F - K-th Largest Triplet

Time Limit: 3 sec / Memory Limit: 1024 MiB

Score : 500 points

Problem Statement

You are given three integer sequences of length N, namely $A = (A_1, A_2, \ldots, A_N)$, $B = (B_1, B_2, \ldots, B_N)$, and $C = (C_1, C_2, \ldots, C_N)$, and an integer K.

For each of the N^3 choices of integers i, j, k $(1 \leq i, j, k \leq N)$, compute the value $A_iB_j + B_jC_k + C_kA_i$. Among all these values, find the K-th largest value.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $1 \leq K \leq \min(N^3, 5 \times 10^5)$
- $1 \leq A_i, B_i, C_i \leq 10^9$
- All input values are integers.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{llll}
N & K & & \\
A_1 & A_2 & \ldots & A_N \\
B_1 & B_2 & \ldots & B_N \\
C_1 & C_2 & \ldots & C_N
\end{array}
$$

Output

Print the answer.

Sample Input 1

```
2 5
1 2
3 4
5 6
```

Sample Output 1

```
31
```

The $N^3 = 8$ values are computed as follows:

- For $(i, j, k) = (1, 1, 1)$: $A_1B_1 + B_1C_1 + C_1A_1 = 1 \times 3 + 3 \times 5 + 5 \times 1 = 23$
- For $(i, j, k) = (1, 1, 2)$: $A_1B_1 + B_1C_2 + C_2A_1 = 1 \times 3 + 3 \times 6 + 6 \times 1 = 27$
- For $(i, j, k) = (1, 2, 1)$: $A_1B_2 + B_2C_1 + C_1A_1 = 1 \times 4 + 4 \times 5 + 5 \times 1 = 29$
- For $(i, j, k) = (1, 2, 2)$: $A_1B_2 + B_2C_2 + C_2A_1 = 1 \times 4 + 4 \times 6 + 6 \times 1 = 34$
- For $(i, j, k) = (2, 1, 1)$: $A_2B_1 + B_1C_1 + C_1A_2 = 2 \times 3 + 3 \times 5 + 5 \times 2 = 31$
- For $(i, j, k) = (2, 1, 2)$: $A_2B_1 + B_1C_2 + C_2A_2 = 2 \times 3 + 3 \times 6 + 6 \times 2 = 36$
- For $(i, j, k) = (2, 2, 1)$: $A_2B_2 + B_2C_1 + C_1A_2 = 2 \times 4 + 4 \times 5 + 5 \times 2 = 38$
- For $(i, j, k) = (2, 2, 2)$: $A_2B_2 + B_2C_2 + C_2A_2 = 2 \times 4 + 4 \times 6 + 6 \times 2 = 44$

Sorting these values in descending order, we have $(44, 38, 36, 34, 31, 29, 27, 23)$, so the 5th largest value is 31.

Sample Input 2

```
3 10
100 100 100
100 100 100
100 100 100
```

Sample Output 2

```
30000
```

Sample Input 3

```
5 54
800516877 573289179 26509423 168629803 696409999
656737335 915059758 201458890 931198638 185928366
140174496 254538849 830992027 305186313 322164559
```

Sample Output 3

```
689589940713840351
```

G - Many LCS

Time Limit: 2 sec / Memory Limit: 1024 MiB

Score : 600 points

Problem Statement

You are given a lowercase English string S of length N and an integer M. For each $k = 0, 1, \ldots, N$, solve the following problem:

- There are 26^M lowercase English strings of length M. Among these, find the number, modulo 998244353, of strings whose longest common subsequence with S has length exactly k.

Constraints

- $1 \leq N \leq 10$
- $1 \leq M \leq 100$
- N and M are integers.
- S is a lowercase English string of length N.

Input

The input is given from Standard Input in the following format:

$$
\begin{array}{l} N \quad M \\ S \end{array}
$$

Output

Let ans_i be the answer for $k = i$. Print the answers in the following format:

$$
\mathrm{ans}_0 \quad \mathrm{ans}_1 \quad \ldots \quad \mathrm{ans}_N
$$

Sample Input 1

```
2 2
ab
```

Sample Output 1

```
576 99 1
```

The answers for $k = 0, 1, 2$ are as follows:

- For $k = 0$: Among length 2 lowercase English strings, those with a longest common subsequence of length 0 with `ab` include strings such as `cd`, `re`, `zz`, totaling 576.
- For $k = 1$: Among length 2 lowercase English strings, those with a longest common subsequence of length 1 with `ab` include strings such as `ac`, `wa`, `ba`, totaling 99.
- For $k = 2$: Among length 2 lowercase English strings, there is 1 string (`ab`) whose longest common subsequence with `ab` has length 2.

Sample Input 2

```
3 4
aaa
```

Sample Output 2

```
390625 62500 3750 101
```

Sample Input 3

```
7 50
atcoder
```

Sample Output 3

```
309810541 226923474 392073062 146769908 221445233 435648037 862664208 238437587
```