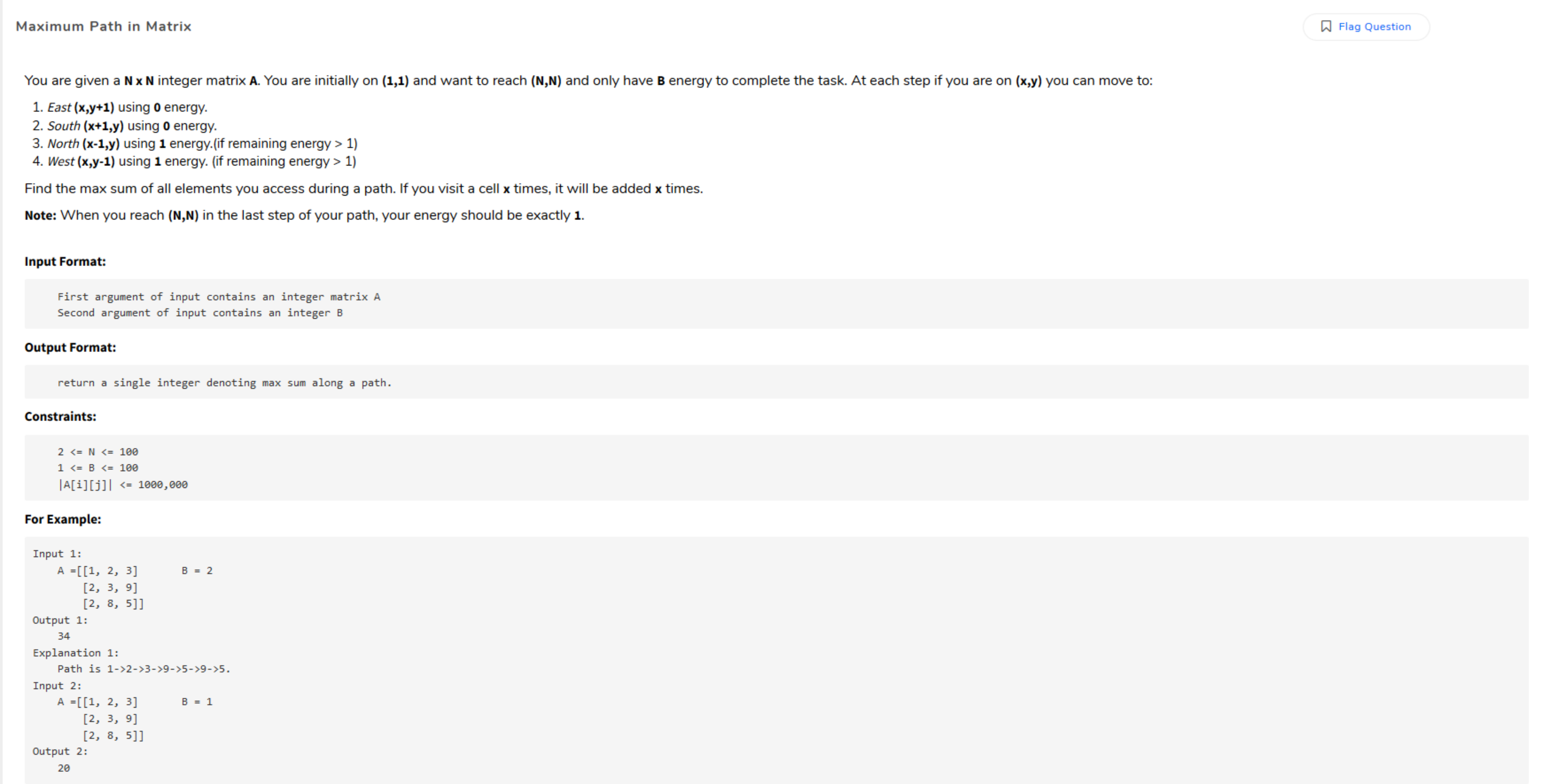

Maximum Path in Matrix

You are given a **N x N** integer matrix **A**. You are initially on **(1,1)** and want to reach **(N,N)** and only have **B** energy to complete the task. At each step if you are on **(x,y)** you can move to:

1. *East* **(x,y+1)** using **0** energy.
2. *South* **(x+1,y)** using **0** energy.
3. *North* **(x-1,y)** using **1** energy.(if remaining energy > 1)
4. *West* **(x,y-1)** using **1** energy. (if remaining energy > 1)

Find the max sum of all elements you access during a path. If you visit a cell **x** times, it will be added **x** times.

Note: When you reach **(N,N)** in the last step of your path, your energy should be exactly **1**.

Input Format:

```
    First argument of input contains an integer matrix A
    Second argument of input contains an integer B
```

Output Format:

```
    return a single integer denoting max sum along a path.
```

Constraints:

```
    2 <= N <= 100
    1 <= B <= 100
    |A[i][j]| <= 1000,000
```

For Example:

```
Input 1:
    A =[[1, 2, 3]       B = 2
        [2, 3, 9]
        [2, 8, 5]]
Output 1:
    34
Explanation 1:
    Path is 1->2->3->9->5->9->5.
Input 2:
    A =[[1, 2, 3]       B = 1
        [2, 3, 9]
        [2, 8, 5]]
Output 2:
    20
```