Problem 1 - Glass Bridge 2.0 (Programming) (12 points)

Problem Description

Improved from the fifth game in *Squid Game (2021)*, Glass Bridge 2.0 is another deadly game played on an $n \times m$ grid, where contestants have to reach the goal cell by jumping from a square glass to another. To better describe the environment, we use (i, j) to denote the cell on the i-th row and on the j-th column. The starting cell is located on $(1, 1)$ while the goal one is located on (n, m). For each of the remaining cells, the cell (i, j) satisfies **exactly one** of the following states:

- The cell is placed with a square glass with a given risk value, since the glass may be smashed when you step on it. To make the game more exciting, there may be a bundle of banknotes on that piece of glass. Considering the revenue and risk, the **expected profit** could be modeled as $a_{i,j}$ if one jumps onto the glass.
- The cell is empty, since a previous contestant has smashed the glass there. In other words, the cell is now impassable – if one steps on this cell, she/he will fall into a horrible abyss immediately.

You, as the next contestant to move on, are going to reach the goal cell by jumping from one cell to another. A jump from (x, y) to (x', y') is valid if **all** of the following conditions hold:

- The cell (x', y') is placed with a square glass.
- $x \leq x' \leq n$
- $y \leq y' \leq m$
- $1 \leq (x' - x) + (y' - y) \leq k$

Currently standing on the starting cell, $(1, 1)$, you would like to first realize if it is possible to reach the goal cell with those currently existing glasses. If so, please come up with a jumping path such that S, the sum of expected profit over all cells on your path, is maximized.

Input

The first line of the input contains 1 integer T – the number of test cases. Then, in each test case:

- The first line is an empty line. This may make it more friendly for you to distinguish between test cases with your naked eyes.
- The second line contains 3 integers n, m, k, denoting the size of the grid and your jumping ability.
- Then, n lines follow, the i-th of which contains m strings or integers, the j-th of which is:
  - 0, if $(i, j) = (1, 1)$ or (n, m).
  - $a_{i,j}$, if the cell (i, j) is placed with a square glass.
  - "`X`" (without quotation marks), if the cell (i, j) is empty.

Constraints

- $1 \leq T \leq 100$
- $1 \leq n, m \leq 10^6$
- $2 \leq nm \leq 10^6$
- $\sum nm \leq 10^6$
- $1 \leq k \leq n + m - 2$
- $-456, 456, 456, 456 \leq a_{i,j} \leq 456, 456, 456, 456$
- You may obtain up to 10.2 points from this problem even if your solution is not fully accepted.

Test Group 0 (0 %)

- Sample Input

Test Group 1 (15 %)

- $n, m \leq 30$
- $k = 1$

Test Group 2 (25 %)

- $n, m \leq 30$

Test Group 3 (45 %)

- $\Big(n, m \leq 30\Big)$ or $\Big(T = 1$ and $n, m \leq 400\Big)$

Test Group 4 (15 %, Bonus Subtask)

- No additional constraint

Output

For each test case:

- In the first line, please output a string s. $s =$ `Passable` if it is possible for you to reach the goal cell with those currently existed glasses. Otherwise, $s =$ `Impassable`.
- If $s =$ `Passable`, you have to specify your jumping path. In the second line, please output S, the maximized sum of the expected profit. In the third line, please output L, the length (including the starting and goal cells) of your path planned. Then L lines follow, the i-th of which should contain 2 integers, x_i, y_i, denoting the i-th coordinate in your path. Note that you **don't** have to minimize the length of your path. If there are multiple solutions, you may print any.

Hint

1. Since each input includes several independent test cases, please carefully clear all results of the current test case before dealing with the next one.
2. In Sample Input 1, the first and second test cases are exactly the same. However, the Sample Output comes up two different paths to answer the same test case. This is just a demonstration on the special scoring method.
3. Test Group 4 may be a challenging subtask. This could be viewed as a bonus subtask since you are still able to obtain 50 points from programming problems even without solving it. Nevertheless, you are encouraged to give it a try because it's solvable without any advanced data structure or algorithm – what you need is just an ingenuity!

Sample Input 1

```
5

3 3 1
0 0 0
-1 0 -1
X -1 0

3 3 1
0 0 0
-1 0 -1
X -1 0

3 3 1
0 -7 X
X X 20
0 10 0

3 3 2
0 -7 X
X X 20
0 13 0

5 7 9
0 X X X X X X
X X X X X X X
X X X -456456456456 X X X
X X X X X X X
X X X X X X 0
```

Sample Output 1

```
Passable
-1
5
1 1
1 2
2 2
3 2
3 3
Passable
-1
5
1 1
1 2
2 2
2 3
3 3
Impassable
Passable
13
4
1 1
1 2
2 3
3 3
Passable
-456456456456
3
1 1
3 4
5 7
```

Behind the Scene

Some wise words from YP, one of the lead TAs, upon knowing the Test Group 4 in this problem: