UVa Online Judge

12066 The Fairy Tale of ICPC

It is now year **3004**, the **ACM ICPC** Programming Contest is still one of the largest and most prestigious Programming Contest of the World. This contest began almost **1030** years ago and many dont remember the history of this contest now. Some say that there is a wonderful story behind this contest. The story goes as follows:

Long long ago there lived a happy dove couple who loved each other very much. They were living peacefully but one day the female dove became very ill. The male dove began roaming around everywhere looking for proper medicine and found a wizard who knew the cure. But the wizard would only give him (male dove) the medicine if he could solve a problem for him, which was in the wizards mind for a long time. As the problem was quite hard the male dove could not solve it. At that time a few very kind men came to his rescue. They gathered all the brilliant programmers from all over the world just to solve the wizards problem, this gathering was known as the first **ACM ICPC World Finals**. The wizards problem was solved and the female dove got cured. Nice story! Isnt it? But the real problem is that now you have to solve this problem on your own.

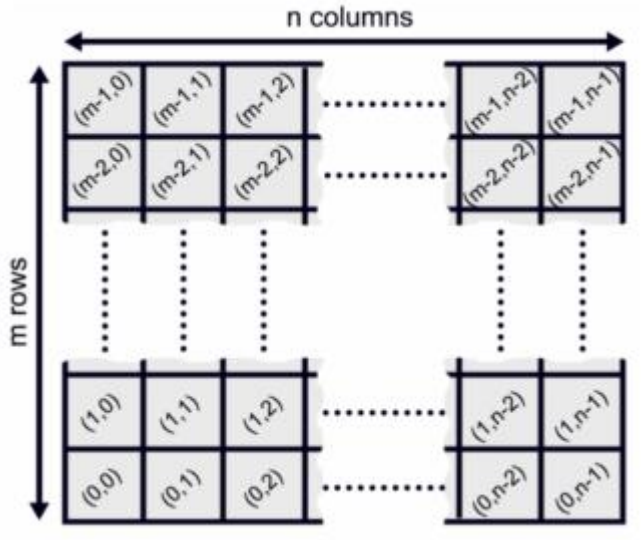

Figure 1

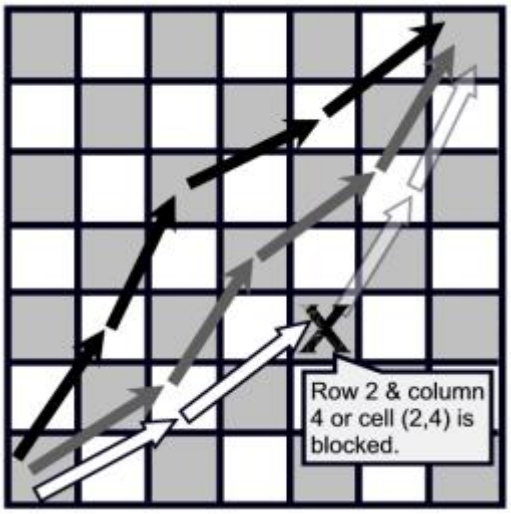

Figure 2

The wizards problem was related to a chessboard. Given an $m \times n$ chessboard (m=number of rows and n=number of columns) one has to go from the lower left corner to the upper right corner. The lower left corner is numbered as $(0, 0)$ and the upper right corner is marked as $(m-1, n-1)$ as shown in **Figure 1**. But in each step one can only jump to another box that is p blocks away in the horizontal direction and q blocks away in the vertical direction or viceversa. No such moves, which would take one away from the destination, are allowed. So if the value $p = 1$ and $q = 3$ and one is in the r-th row and c-th column or location (r, c) then he can only jump to location $(r + 1, c + 3)$ or $(r + 3, c + 1)$ but he cannot jump to locations like $(r - 1, c + 1)$ or $(r - 1, c - 1)$, etc. In **Figure 2** we can see a situation where $p = 2$ and $q = 1$ (or $p = 1$ and $q = 2$). It shows two of the total six possible ways (considering

that no square is blocked) of going from the lower left corner to the upper right corner and each of these two ways requires four steps to reach the destination. It also shows that the square at $(2, 4)$ (row **2**, column **4**) is blocked, which has ruled out the possibility of a path through that square. Given the value of m, n, p and q and position of the squares which are blocked, your job is to find out the number of steps one requires to reach the destination and also the total number of possible ways to reach the destination.

Input

The input file contains less than **100** sets of inputs. The description of each set is given below:

Each line contains five integers m, n $(6 < m, n < 4001)$, p, q $(0 < p, q < 10)$ and b $(0 \leq b \leq 10)$. The meaning of m, n, p and q are described in the problem statement above. The integer b denotes the number of squares that are blocked. Each of the next b lines contains two integers (r_i, c_i) which denotes the row and column of the i-th blocked square.

Input is terminated by a set where the value of $m = n = 0$. This set should not be processed. Please note that at least **90%** of the input test cases follow the limit $(6 < m, n < 300)$.

Output

For each set of input produce one line of output, which contains two integers C and W. Here C is the number of steps required to go from the lower left corner to the upper right corner and W is the total possible ways one can go from lower left corner to the upper right corner. If it is impossible to go from the lower left corner to the upper right corner with the given configuration then print a line '`Impossible`' instead as shown in the output for sample input. W can have at most **802** digits.

Sample Input

```
8 7 1 2 2
2 1
1 2
8 8 1 3 0
49 49 2 1 0
93 109 3 1 1
24 16
108 50 3 1 6
85 47
45 23
54 34
24 40
40 24
82 46
0 0 0 0 0
```

Sample Output

```
Case 1: Impossible
Case 2: Impossible
Case 3: 32 601080390
Case 4: 50 64542614482000
Case 5: 39 376817
```