The Lord of the Kings

`king.c`, `king.cpp`, `King.java`, `king.py`

After many long and exhausting years of war with dangerous overseas nations, our country has finally managed to get rid of almost all opposing forces and emerged victorious. Such a glorious victory shall be remembered and celebrated for many upcoming years. Because of that, our king has decided to declare the day of the victory as a public holiday, when a victory parade is to be held. During the parade, the king accompanied by his troops will start from his palace and subsequently visit all cities in the country.

The king and his suite will travel by a new type of environment-friendly electric helicopter that has a disadvantage of a relatively short operating radius. The king has requested you and your advisors to build helipads on some farm fields and in all the cities, so that each city is reachable from his palace by a sequence of short flights between helipads. However, building the helipads and the supporting infrastructure is costly. It is thus important to minimize the number of farm fields on which helipad construction takes place.

In addition, due to specific helicopter design, the king and his troops need to move in a special pattern which may influence the number and the locations of helipads.

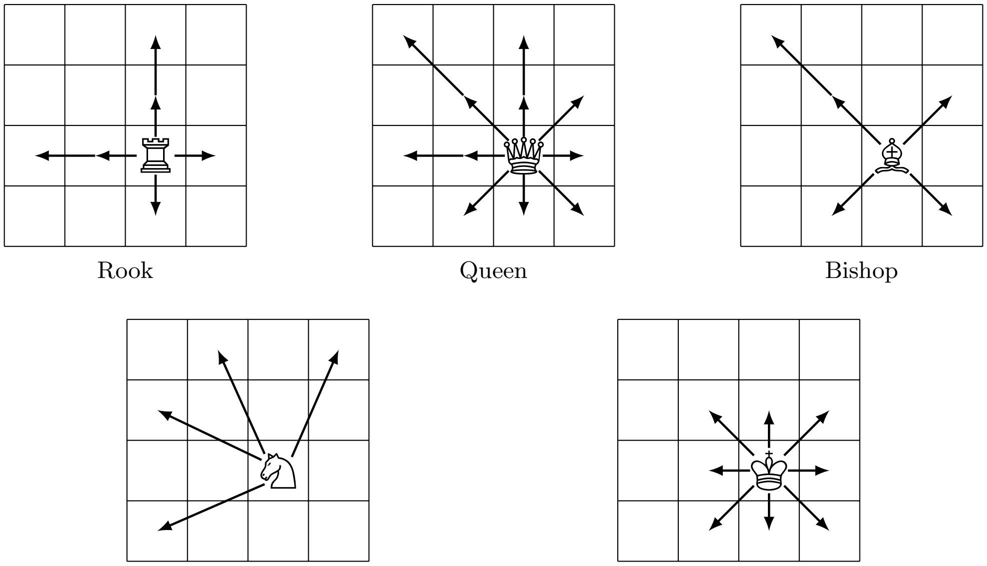
You are given a rectangular grid map of the country, which consists of tiles of farm fields, tiles of cities and a tile of the king's palace. Also, you are given the movement pattern of the helicopter — it may move either as a Rook, a Queen, a Bishop, a Knight, or as a King in the game of chess (see images for movement illustration). Your task is to determine the minimal number of farm fields tiles and city tiles which have to host a helipad, in order to fulfill the conditions specified above. The tile with the king's palace already has a helipad and does not need a new one.

Input Specification

The first line of input contains two integers N and M ($1 \leq N, M \leq 15$), where N is the number of rows and M is the number of columns in the grid representing our country. The second line of input contains two integers X and Y ($1 \leq X \leq N$, $1 \leq Y \leq M$), representing the position of the king's palace, followed by a single character determining the movement pattern ("`R`" – Rook, "`Q`" – Queen, "`B`" – Bishop, "`N`" – Knight, "`K`" – King). The third line of input contains an integer T ($1 \leq T \leq 10$) representing the number of cities in the country. After that, T lines follow, each containing two integers W and Z ($1 \leq W \leq N$, $1 \leq Z \leq M$). Each such line represents a position of one city tile. All cities occupy different tiles, no city occupies the palace tile. All tiles that do not represent a city or palace are considered to be farm fields.

Output Specification

Output a single integer — the minimal number of farm/city tiles to host a helipad. Should it be impossible for the king and his troops to visit all cities, output -1.



Rook

Queen

Bishop

Knight

King

Figure 1: Illustration of specific helicopter movement patterns.

Sample Input 1

```
3 3
3 1 K
2
1 1
1 3
```

Output for Sample Input 1

```
3
```

Sample Input 2

```
3 3
3 1 Q
2
1 1
1 3
```

Output for Sample Input 2

```
2
```

Sample Input 3

```
5 5
4 4 R
4
1 2
2 1
2 5
5 1
```

Output for Sample Input 3

```
6
```