Senior Software Engineer Test

1. Fraudulent Activity Notifications

HackerLand National Bank has a simple policy for warning clients about possible fraudulent account activity. If the amount spent by a client on a particular day is *greater than or equal to* **2** * the client's median spending for a trailing number of days, they send the client a notification about potential fraud. The bank doesn't send the client any notifications until they have at least that trailing number of prior days' transaction data.

Given the number of trailing days and a client's total daily expenditures for a period of days, determine the number of times the client will receive a notification over all days.

Example

$expenditure = [10, 20, 30, 40, 50]$
$d = 3$

On the first three days, they just collect spending data. At day **4**, trailing expenditures are **[10, 20, 30]**. The median is **20** and the day's expenditure is **40**. Because **40 >= 20 * 2**, there will be a notice. The next day, trailing expenditures are **[20, 30, 40]** and the expenditures are **50** . This is less than **2 * 30** so no notice will be sent. Over the period, there was one notice sent.

Input Format
The first line contains two space-separated integers ***n*** and ***d***, the number of days of transaction data, and the number of trailing days' data used to calculate median spending respectively.
The second line contains ***n*** space-separated non-negative integers where each integer ***i*** denotes ***expenditure[i]*** .

Constraints

- $1 \leq n \leq 2 \times 10^5$
- $1 \leq d \leq n$
- $0 \leq expenditure[i] \leq 200$

Output Format

Sample Input 0

```
STDIN               Function
-----               --------
9 5                 expenditure[] size n =9, d = 5
2 3 4 2 3 6 8 4 5   expenditure = [2, 3, 4, 2, 3, 6, 8, 4, 5]
```

Sample Output 0

```
2
```

Sample Input 1

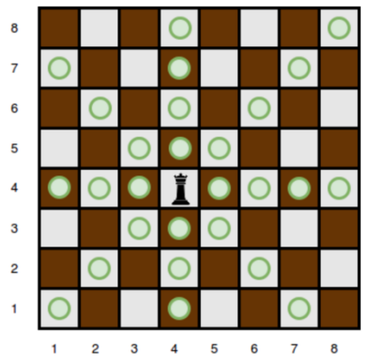
```
5 4
1 2 3 4 4
```

Sample Output 1

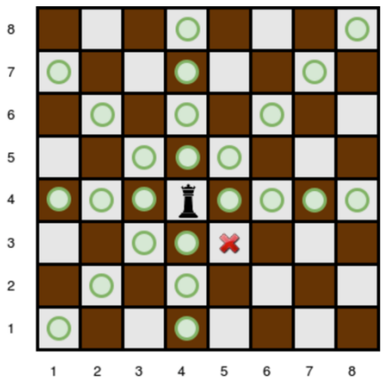
```
0
```

2. Queen's Attack

You will be given a square chess board with one queen and a number of obstacles placed on it. Determine how many squares the queen can attack.
A queen is standing on an ***n x n*** chessboard. The chess board's rows are numbered from ***1*** to ***n***, going from bottom to top. Its columns are numbered from ***1*** to ***n*** , going from left to right. Each square is referenced by a tuple, ***(r,c)***, describing the row, ***r***, and column, ***c***, where the square is located.
The queen is standing at position ***(r_q,c_q)***. In a single move, she can attack any square in any of the eight directions (left, right, up, down, and the four diagonals). In the diagram below, the green circles denote all the cells the queen can attack from ***(4, 4)*** :



There are obstacles on the chessboard, each preventing the queen from attacking any square beyond it on that path. For example, an obstacle at location ***(3,5)*** in the diagram above prevents the queen from attacking cells ***(3,5)***, ***(2,6)***, and ***(1,7)*** :



Given the queen's position and the locations of all the obstacles, find and print the number of squares the queen can attack from her position at ***(r_q,c_q)***. In the board above, there are **24** such squares.

Description

- *int n:* the number of rows and columns in the board
- i*nt k:* the number of obstacles on the board
- *int r_q:* the row number of the queen's position
- *int c_q:* the column number of the queen's position
- *int obstacles[k][2]:* each element is an array of **2** integers, the row and column of an obstacle

Returns
- *int:* the number of squares the queen can attack

Constraints

- $0 < n \leq 10^5$
- $0 \leq k \leq 10^5$
- A single cell may contain more than one obstacle.
- There will never be an obstacle at the position where the queen is located.

Sample Input 1

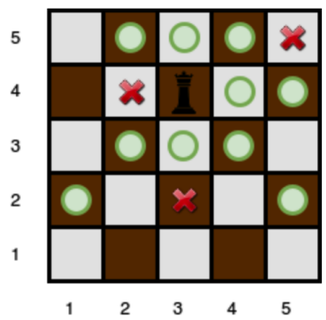
```
5 3
4 3
5 5
4 2
2 3
```

Sample Output 1

```
10
```

Explanation 1

The queen is standing at position $(4, 3)$ on a 5×5 chessboard with $k = 3$ obstacles:



The number of squares she can attack from that position is 10.

Note :
Please use one of these programming languages to solve the problems:

- C
- C++
- PHP
- Python
- Go

Submit your answer on github and send your github link to this email: hr@perqara.com