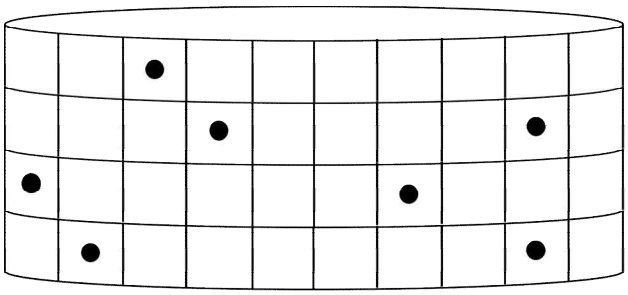
계속되는 알고리즘 공부에 잠시 지친 경근이는 다른 세계에서 몇 일간 쉬고 오면 좋을 것이라는 생각을 했다. 그러던 어느 날, 다른 세계로 떠나는 문이 있는 장소를 상수에게 듣게 되어 그 곳으로 갔다. 그 곳에는 다른 세계로 떠나는 문이 있었는데 잠겨 있었고, 문을 열 수 있는 커다란 다이얼 식 자물쇠가 있었다.

자물쇠는 위아래로 M개의 다이얼로 만들어져 있다. 각 다이얼은 R개의 칸이 있고 좌우로 칸에 맞추어 돌릴 수 있다. 다이얼의 각 칸에는 점이 있을 수도 있고 점이 없을 수도 있다. 각 다이얼에는 최소 1개의 점이 있다. 다이얼들을 돌려서 어떤 위치에서든 **세로로 모든 칸에 점이 있도록** 만들면 문이 열린다고 한다. 각각의 다이얼은 따로 돌아간다. 다이얼이 아주 무거워 돌리기 힘들기 때문에 다이얼을 돌리는 칸 수의 합을 최소화하고 싶다.



위 그림은 다이얼의 한 예를 보여 준다. 가능한 최선의 방법 중 하나는, 위에서 첫 번째 다이얼을 왼쪽으로 한 칸, 두 번째 다이얼을 왼쪽으로 두 칸, 세 번째 다이얼을 오른쪽으로 한 칸, 네 번째 다이얼을 돌리지 않아서, 총 4칸을 돌리는 방법이다.

다이얼의 크기와 점들의 위치를 입력으로 받아서 문이 열리도록 최소 칸 수로 다이얼을 돌리는 방법을 알아내는 프로그램을 작성하라. 점들의 위치는 다이얼 번호와 칸 번호의 쌍인 좌표로 주어진다. 제일 위의 다이얼이 0번 다이얼이다. 아래로 가면서 다이얼 번호가 1씩 증가한다. 제일 아래 다이얼이 $M-1$번 다이얼이 될 것이다. 칸 번호를 정하기 위해 임의의 동일한 위치의 (즉, 세로로 한 줄에 위치한) 칸들을 0번으로 잡는다. 오른쪽으로 가면서 칸 번호가 1씩 증가한다. 0번 칸의 왼쪽에는 $R-1$번 칸이 있을 것이다. 주어진 점들의 좌표가 겹치는 경우는 없다.

- long long findMinClicks(int M, int R, vector< pair<int, int> > P) : M은 다이얼의 개수이다. R은 한 다이얼의 칸 수이다. P는 크기 N인 int 쌍들의 배열이다. 한 쌍의 첫 자연수는 다이얼 번호, 두 번째 자연수는 칸 번호이다. 주어진 점들의 좌표가 겹치는 경우는 없다. findMinClicks() 함수는 문이 열리도록 최소 칸 수로 다이얼을 돌리는 방법에서 돌리는 칸 수를 리턴해야 한다.

구현 세부사항

여러분은 dial.cpp라는 이름의 정확히 하나의 파일을 제출해야 한다. 이 파일에는 다음의 함수가 구현되어 있어야 한다.

- long long findMinClicks(int M, int R, vector< pair<int, int> > P)

이 함수는 위에서 설명한 것과 같이 동작하여야 한다. 물론 다른 함수들을 만들어서 내부적으로 사용할 수 있다. 제출한 코드는 입출력을 수행하거나 다른 파일에 접근하여서는 안된다.

grader 예시

주어지는 grader는 다음과 같은 형식으로 입력을 받는다.

- line 1: 정수 N, M, R
- line 2 ~ $N+1$: 정수 D_i, C_i

위에서 (D_i, C_i)는 i번째 점의 좌표이다. 주어진 grader는 여러분의 코드가 findMinClicks() 함수에서 리턴한 값을 한 줄에 출력한다.

제한 조건

- $1 \leq R \leq 10^9$, $1 \leq N \leq \min(MR, 500{,}000)$, $1 \leq M \leq N$.

서브태스크 1 [10 points]

- $1 \leq N \leq 5{,}000$, $1 \leq R \leq 5{,}000$

서브태스크 2 [15 points]

- $1 \leq M \leq 5{,}000$, $1 \leq R \leq 5{,}000$

서브태스크 3 [16 points]

- $1 \leq N \leq 5{,}000$

서브태스크 4 [109 points]

- **원래 제한 조건 이외의 추가적인 제한 조건이 없음**

[입력 예]

```
7 4 20
1 3
0 2
2 6
3 8
3 1
2 0
1 8
```

위의 입력에서 여러분의 코드가 동작하는 방식에 따른 실행 예를 아래에 보인다.

호출	리턴 값
▪ findMinClicks(7, 4, 20, { {1, 3}, {0, 2}, {2, 6}, {3, 8}, {3, 1}, {2, 0}, {1, 8} })	4