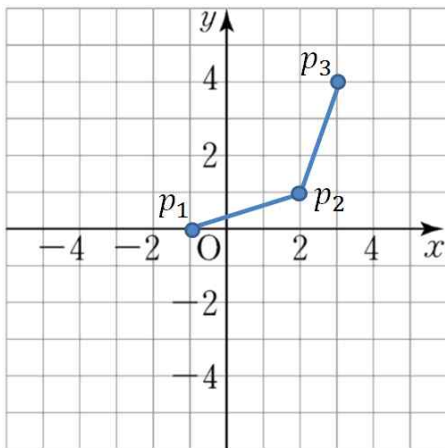


좌회전/우회전

x, y -평면상에 있는 서로 다른 두 점 p_1, p_2 에 대하여 이 두 점을 잇는 선분을 $\overline{p_1p_2}$ 로 나타낸다. 문제는 점 p_2 를 공유하는 연속적인 두 선분 $\overline{p_1p_2}$ 와 $\overline{p_2p_3}$ 이 있을 때, 선분을 따라 가면서 점 p_1, p_2, p_3 를 순서대로 지나가려면 점 p_2 에서 좌회전하는지 아니면 우회전하는지를 결정하는 프로그램을 작성하는 것이다. p_2 에서 좌회전도 아니고 우회전도 아닌 경우도 있는데, 바로 세 점 p_1, p_2, p_3 가 일직선상에 놓여 있는 경우이다.

예를 들어 아래 그림과 같이 $p_1 = (-1, 0)$, $p_2 = (2, 1)$, $p_3 = (3, 4)$ 라면 p_1 에서 출발하여 p_2 에 도달한 다음 p_2 에서 좌회전하여 p_3 에 이르게 된다. 또한 p_1, p_2 는 같은데 $p_3 = (4, 0)$ 이라면 우회전할 것이고, $p_3 = (5, 2)$ 라면 세 점이 일직선상에 위치하게 될 것이다.



프로그램 이름은 turn.cpp(c), 설명 파일 이름은 turn.pdf로 하고, 프로그램의 실행시간은 1.0초를 초과할 수 없다. 부분 점수는 없다.

입력 형식

표준 입력을 통하여 입력한다. 첫째 줄에 점 p_1 의 x, y -좌표가, 둘째 줄과 셋째 줄에 각각

점 p_2 와 p_3 의 좌표가 입력된다. 각 점의 좌표 값은 $-10,000$ 이상 $10,000$ 이하인 정수이다. 입력으로 주어진 세 점이 모두 서로 다른 위치에 놓여 있다고 가정하여도 좋다.

출력 형식

표준 출력을 통하여 출력한다. 연속된 두 선분 $\overline{p_1p_2}$ 와 $\overline{p_2p_3}$ 을 따라갈 때 점 p_2 에서 좌회전하면 L, 우회전하면 R, 세 점이 일직선상에 위치하면 C를 한 줄에 출력한다.

입력과 출력의 예 (1)

입력

```
-1 0
2 1
3 4
```

출력

```
L
```

입력과 출력의 예 (2)

입력

```
-1 0
2 1
4 0
```

출력

```
R
```

입력과 출력의 예 (3)

입력

```
-1 0
2 1
-4 -1
```

출력

```
C
```

※ L은 left-turn, R은 right-turn, 그리고 C는 collinear를 의미한다.