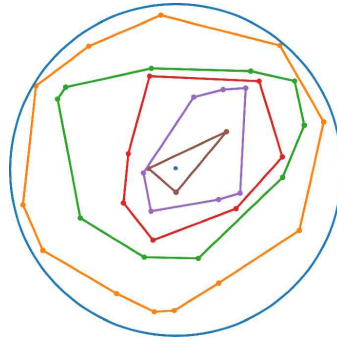




회전 정렬

[문제] 2차원 공간에 N 개의 서로 다른 정수 점 $P_i = (x_i, y_i)$ 이 있다. 단 $x_i, y_i \in \mathbb{Z}^+$ 이다. 이 점의 집합을 “특별한” 방법으로 정렬하고자 한다. 우리가 시행하는 정렬은 x 축으로의 무한 직선 $[0, +\infty]$ 을 반시계 방향으로 회전할 때 만나는 점의 순서로 결정된다. 만일 하나의 점 P_r 이 선택되면 그 다음에는 이 점을 회전축으로 다시 직선을 또 다른 점을 만날 때까지 회전한다. 그리고 한번 선택된 점은 더 이상 고려되지는 않는다. 아래 13개의 점 집합을 예를 이용해서 설명해 보자. 시작 점은 항상 $P_0 = (0,0)$ 으로 고정되어 있다.

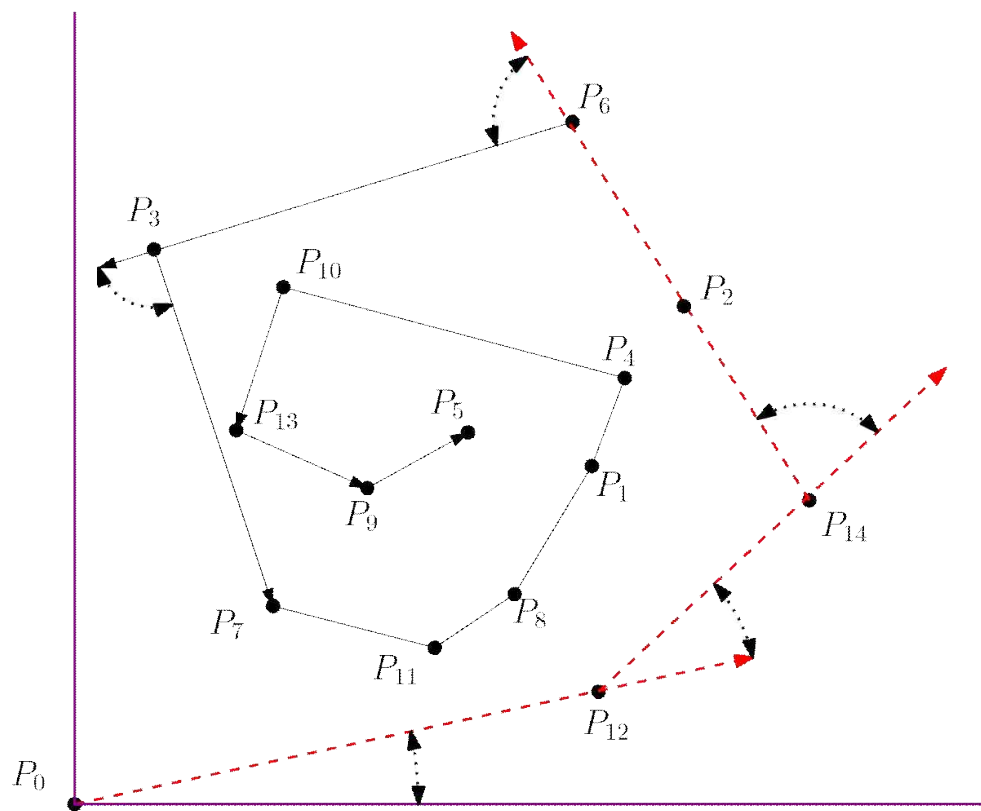


그림-1. 13개의 점을 winding(감기) 기준으로 정렬하기. 위 점들의 회전 정렬 순서는 P_i 의 index i 로 볼 때 [12, 14, 2, 6, 3, 7, 11, 8, 1, 4, 10, 13, 9, 5]이다.

[입출력] 입력 파일 **stdin** 의 첫 줄에 점의 개수 N 이 주어진다. 단 $3 \leq N \leq 30$ 이다. 이어지는 N 개의 줄에는 P_i 의 좌표가 index i 와 함께 ' $i \ x_i \ y_i$ ' 3개의 정수로 주어진다. 여러분은 $i = 1, 2, 3, \dots, N$ 이다. 여러분은 $P_0 = (0, 0)$ 부터 시작하는 감기 정렬의 정점 번호를 한 줄에 하나씩 N 개의 줄에 출력해야 한다. 만일 두 점이 P_2 와 P_6 과 같이 직선 상에 정확하게 존재한다면 거리가 짧은 점을 더 빠른 순서로 선택한다. 즉 [14, 2, 6]이 되어야 한다.

stdin	stdout
14 // N	12
1 13 9	14
2 15 13	2
3 2 14	6
4 14 11	3
5 10 10	7
6 13 17	11
7 5 5	8
8 11 5	1
9 7 8	4
10 6 13	10
11 9 4	13
12 13 3	9
13 4 9	5
14 18 7	

[제한조건] 프로그램 이름은 **wsort.{c,cpp,java,py}**이며 제출 허용 횟수는 25회이다. 데이터 당 제한시간은 1초, 그리고 token은 최대 700개이다. 단 이번 과제부터는 **#define** 문자열을 사용하여 코드의 길이를 줄이는 편법은 사용할 수 없다. **#define**은 반드시 1줄만 가능하며 **multiline string**은 더 이상 허용하지 않는다. 만일 이 방법을 사용하면 이전 점수에 관계없이 0점 처리된다.

[제한조건] 점집합 S 가 있을 때 여기에 속한 어떤 2점 (p, q) 를 선택한다. 이 둘을 연결한 support line을 구성한 다음, $w \in S \setminus \{p, q\}$ 의 모든 점 w 가 한쪽으로 몰려있다면, 즉 $\text{signed_area}(p, q, w)$ 가 같은 부호 이라면 이 두 점은 sorting 점이 된다.