

에어로배틱스

(Output Only Task)

비타로는 비행기 대회에 출전할 예정이다. 대회에서 비타로는 비행기를 조종한다. 비행기는 특정한 높이를 날면서 지정된 위치들을 지나가야 한다. 비행기는 좌표평면 위를 날아가는 것으로 생각하자. 지정된 위치는 N 개가 있고, 1번부터 N 번까지 번호가 붙어 있다. 번호가 i 인 ($1 \leq i \leq N$) 지정된 위치의 좌표는 (X_i, Y_i) 이다.

대회 동안 비행기는 각 지정된 위치를 한번씩 지나야 한다. 구체적으로, 다음의 규칙을 따라야 한다.

1. 비타로는 우선 지정된 위치 중 하나를 시작 위치로 잡는다. 비행기는 그 위치에서 출발한다.
2. 다음을 $N - 1$ 번 반복한다.

아직 선택되지 않은 지정된 위치들 중 하나를 다음번 지정된 위치로 선택한다. 비행기는 현재 위치에서 그 위치로 똑바로 날아간다.

3. 마지막 지정된 위치에 도착했을 때 모든 비행은 종료된다.

위의 2번 단계에서 시작 위치는 이미 선택된 것으로 간주한다. 비행기는 한 지정된 위치에서 다른 지정된 위치로 똑바로 날아가야 한다. 즉, 선분들이나 곡선을 따르면 안된다.

비행기가 날아가는 궤적은 연속된 선분들로 이루어진다. 비행기가 날아가는 동안 방향을 바꾸는 것을 최대 $N - 2$ 번 하게 된다. 한 지정된 점에서 두 선분이 이루는 각도가 작으면, 즉, 비행기가 방향을 바꾸는 각도가 크다면, 비행이 불안정하게 될 수 있다.

그래서, 비타로는 시작과 마지막을 제외한 $N - 2$ 개의 지정된 위치에서 선분이 만나는 각도들 중 최소값을 가장 크게 하고 싶다.

지정된 위치들의 좌표를 받아서 선분들이 만나는 각도의 최솟값이 가장 크게 되도록 지정된 위치들의 순서를 배열하는 프로그램을 작성하라.

Input

입력은 다음 형식으로 주어진다. 모든 값들은 정수이다. Z_0 는 그레이더가 사용하는 값이다.

```

N Z0
X1 Y1
⋮
XN YN

```

Output

출력은 N 개의 줄로 되어 있어야 한다. 출력 중 k 번째 ($1 \leq k \leq N$) 줄에는 P_k ($1 \leq P_k \leq N$)가 출력되어야 한다. P_k 는 비행기 궤적의 k 번째 지정된 위치이다. 비행기가 출발하는 위치는 P_1 이다.

Submission

각각의 입력 파일 `input_01.txt`, `input_02.txt`, ..., `input_06.txt`에 대해 출력 파일 `output_01.txt`, `output_02.txt`, ..., `output_06.txt`을 제출하라.

Constraints

- $3 \leq N \leq 1000$.
- $\sqrt{X_i^2 + Y_i^2} \leq 10\,000\,000$ ($1 \leq i \leq N$).
- $(X_i, Y_i) \neq (X_j, Y_j)$ ($1 \leq i < j \leq N$).
- $1 \leq Z_0 \leq 179$.

Library

이 문제에서 3개 점에 대한 각도를 계산해 주는 라이브러리를 사용할 수 있다. 라이브러리는 패키지의 `aerobatics.h`에 있다. 사용 방법은 아래와 같다.

- `double GetAngle(int xa, int ya, int xb, int yb, int xc, int yc)`
이 함수는 각 BAC 를 "도" 단위로 계산한다. 실수 오차는 충분히 작다는 것이 보장된다. **인자들의 순서에 주의하라.**
 - 인자 `xa`와 `ya`는 각각 A 의 x 와 y 좌표 값이다.
 - 인자 `xb`와 `yb`는 각각 B 의 x 와 y 좌표 값이다.
 - 인자 `xc`와 `yc`는 각각 C 의 x 와 y 좌표 값이다.
 - 점 A 와 B 의 좌표가 같거나, 점 A 와 C 의 좌표가 같은 경우 이 함수의 동작은 예상할 수 없다.

이 문제를 푸는 프로그램에서 라이브러리의 `GetAngle` 함수를 사용할 수 있다. 사용하면서 함수를 고치는 것도 허용된다.

`GetAngle` 함수는 그레이더가 사용하는 것과 동일한 함수이다.

Grading

각 출력에 대해 다음과 같이 점수가 계산된다.

답이 틀렸다면 점수는 0점이다. 예를 들어, P_1, P_2, \dots, P_N 가 지정된 점들의 순열이 아니거나 출력 양식이 잘못된 경우 등은 답이 틀린 것이다.

답이 맞다면 점수는 아래와 같이 계산된다. 각도 Z 가 출력 순열의 $N - 2$ 개의 지정된 위치에서 선분들이 만나는 각도의 최소값이라고 하자. 각 케이스에 대한 점수는 배정된 점수 S 에 따라 다음 공식으로 계산된다.

- $Z \geq Z_0$ 이면 점수는 S 이다.
- $Z < Z_0$ 이면 점수는 $S \times \frac{f(Z/180)}{f(Z_0/180)}$ 이다.

함수 $f(\alpha) (0 \leq \alpha \leq 1)$ 는 아래와 같다.

$$f(\alpha) = 4\alpha^4 + \alpha.$$

이 문제에 대한 총 점수는 각 케이스에 대한 점수의 합이다. 점수는 정수로 반올림된다.

각 케이스에 대한 N, Z_0 , 및 배정된 점수는 아래 표와 같다.

Subtask	Input data	N	Z_0	score
1	input_01.txt	15	100	10
2	input_02.txt	200	143	15
3	input_03.txt	200	134	15
4	input_04.txt	1000	156	20
5	input_05.txt	1000	150	20
6	input_06.txt	1000	153	20

Sample Input and Output

표준 입력(stdin)	표준 출력(stdout)
7 90	5
3 1	3
2 5	1
0 2	7
-1 6	6
-3 1	4
-1 -4	2
4 -2	

비행기가 지정된 위치들을 5,3,1,7,6,4,2번의 순서로 지나간다면 비행기의 궤적은 다음 그림과 같다. 선분이 만나는 각도가 가장 작은 지정된 위치는 6번이다. 각도는 $68.19859\dots$ (도)이다. $Z_0 = 90$ (도)이므로 케이스에 배당된 점수의 61.5 %를 받게 된다.

