

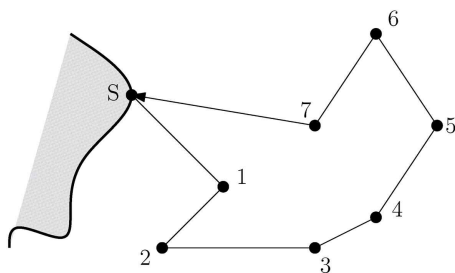


요트(Yacht)

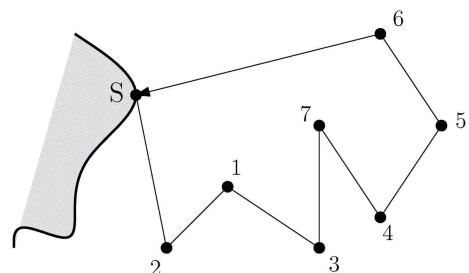
[문제] 동력 장치를 사용하지 않고 오로지 바닷 바람만 이용해서 움직이는 요트(yacht)가 있다.¹⁾ 이 요트는 아침 정해진 항구에서 출발하여 이미 설치된 여러 곳의 통발에 들어간 고기를 수집하여 저녁 무렵에 다시 출발한 항구로 되돌아온다. 즉 요트는 바닷가에 부표로 표시된 특정 지점(통발이 설치된)을 모두 방문한 뒤 되돌아 와야 한다. 그 다음 잡은 고기는 모두 들어내고 통발 속의 미끼는 새로운 미끼로 교체하는 일을 한다.

단 이 요트는 오전에는 해변에서 바닷가 쪽으로, 즉 x 축 기준으로 앞으로(오른쪽)으로만 움직이고 오후에는 바람의 영향으로 반대 방향, 바닷가에서 해변 쪽으로만, 즉 x 축 기준으로 왼쪽으로만 움직일 수 있다. 아래 그림 왼쪽과 같이 그림-1 (a)와 같이 지점-1번을 방문한 뒤에 다시 지점-2번으로 뒤로 갔다가 다시 앞 지점-3번으로 나가는 방식의 이동은 불가능하다.

요트는 그림-1 (b)와 같이 아침에 오른쪽으로 계속 나아가다가 제일 끝을 찍은 그 다음, 돌아올 때에는 왼쪽으로만 이동하며 이전에 방문하지 않은 점을 모두 거쳐서 다시 최초의 시작점 S로 돌아와야 한다. 즉 시작해서 오른쪽으로 쪽- 한번, 그 마지막 점에서 왼쪽으로 쪽 한번, 이렇게 이동하여 모든 점을 빠짐없이 방문한 뒤 다시 처음 지점으로 되돌아 와야 한다.



(a) 1→2는 불가능한 진행



(b) 가능한 진행

그림-1: S점에서 시작하여 요트를 몰아서 다시 S로 돌아오는 경로(tour)의 예

여러분은 이런 제약조건으로 요트를 몰아나갈 때 모든 통발을 수거한 뒤에 방문하고 돌아오는 길 중에서 가장 짧은 경로(shortest tour)를 찾아서 그 길이를 출력해야 한다. 각 어망의 위치 (x,y) 는 정수 점이다. 그리고 두 점 (x,y) 와 (p,q) 의 2차원 거리는 실제 거리의 그 정수 부분만 사용된다. 즉 그 실제 거리가 $\sqrt{125}$ 이라면 $\sqrt{125}=11.18$ 이므로 계산에서는 11로 계산된다. 단 이 문제에서 시작 점 S는 $(0,500)$ 으로 고정되어 있다.

1) 신기하게도 요트는 맞바람이 불어도 지그재그로 움직이며 나아갈 수 있다. 이것이 요트 항해의 핵심이다.

[입출력] 입력과 출력은 모두 표준 입출력을 사용한다. 입력 화일 첫 줄에 정수 N ($3 < N < 100$)이 주어진다. 요트가 출발하는 항수 S 의 좌표는 $S = (0, 500)$ 으로 고정되어 있다. 여러분은 항상 이 점 $(0, 500)$ 에서 출발해야 한다. 그 다음 이어지는 N 개의 줄에는 각 점의 정수 좌표 (x_i, y_i) 가 x_i y_i 로 주어진다. 각 좌표의 범위는 $1 \leq x, y \leq 1,000$ 이다. 여러분은 S 에서 출발하여 모든 통발을 수거하고 다시 S 로 되돌아오는 경로 중에서 가장 짧은 경로를 찾아서 그 길이를 정수로 출력해야 한다. 단 앞서 말했든지 두 통발 사이의 거리는 실제 2차원 거리의 정수 부분만으로 계산한다.

[예제]

stdio	stdout
7 // 7개의 통발 34 123 56 45 340 65 100 300 788 39 // 끝 점. 반환점 407 90 205 654	2610
15 // 15개의 통발 37 159 105 21 76 20 27 67 29 38 108 31 149 116 127 143 122 72 52 118 176 174 // 끝 점. 반환점 97 192 156 141 15 152 120 15	1328

[제한조건] 프로그램 이름은 **yacht.{c, cpp, py}**이며 수행시간은 최대 1초, token은 600개까지 가능.

[도움말] 어떤 경로를 선택하든 '끝'을 찍고 방향을 바꾸는 마지막 점은 x 축 기준으로 가장 큰 값의 점이 되어야 한다. 이 점을 제외한 $N-1$ 개의 통발이 배치된 상황의 답을 모두 알고 있다고 가정하고 오른쪽 끝에 1개의 점을 추가되었을 경우를 고려해야 한다.

$N=1$ 이라면 다른 선택의 여지가 없기 때문에 S 에서 출발하기 그 곳을 찍고 바로 돌아와야 한다. 즉 이 문제는 trivial하다. $N=2$ 라면 가능한 경로는 삼각형 하나 뿐이므로 이것 역시 간단하게 해결된다. $N=3$, $[s, a, b, t]$ 일 때를 생각해보자. 이 순서는 x 축 기준으로 정렬된 것이다. 따라서 return point는 오른쪽 끝 점 t 가 된다. 이 경우 가능한 경로는 $s-a-b-t-s$ 이거나 $s-a-t-b-s$ 이거나 $s-t-b-a-s$ 경우가 가능하다. 즉 t 에 도착하기 바로 이전의 점은 $\{s, a, b\}$ 가 가능하며 t 방문 후 거쳐가는 지점은 $\{s, b, s\}$ 가 가능하다. 만일 $N=1, 2, 3$ 의 답을 모두 알고 있다고 가정할 때 $N=4$ 의 경우는 쉽게 해결된다. 이러한 방법으로 $k-1$ 개가 해결된 상황에서 추가 1개의 지점(가장 오른쪽)이 추가된 경우를 푸는 방식으로 접근해야 한다.