5866 Robot Arm Planning

Your job is to design a robot arm as in Figure 2. However, the control of a robot arm can be complicated, so you want to experiment it on a 2D simplified version (as in Fig. 2) before a real one is built.

A robot can have several joins (or "torques" in their formal name) and each join is controlled by a motor. The motor can only rotate exactly 45 degree clockwise or counter clockwise in one move. For example, to rotate 90 degree, you need two moves. To rotate a join in one move, the motor consumes a fixed amount of power. The goal of robot arm planning is to move the arm to a destination with minimum number of moves of all motors. In Fig. 2(a), a robot arm with 3 arms and 4 joins is shown. Let Join 0 be always fixed at coordinate (0,0). Each arm is 100 cm long and a join has radius of 10 cm. Given a rectangular area defined by (x_1, y_1, x_2, y_2), where (x_1, y_1) is the coordinate of bottom-left corner and (x_2, y_2) is the coordinate of top-right corner, and given a robot arm with N arms and $N + 1$ joins, please compute the minimum moves needed to move join N (the top join) inside that area. For example, in Fig. 2(b), to move the join 3 inside the rectangular area $(100, 200, 200, 300)$ can be simply done by rotating join 1 by one clockwise move. Note that, the body of the top join must be completely enclosed by the rectangular area. In addition, please assume the arms will not block each other. For example, it is valid to have join 1 rotate 180 degree to overlap arm 0 and arm 1.

Figure 1: An robot arm.

Technical Specification

1. $2 \leq N \leq 10$.
2. $-N \times 100 \leq x_1, y_1, x_2, y_2 \leq N \times 100$, where x_i's and y_j's are integers.

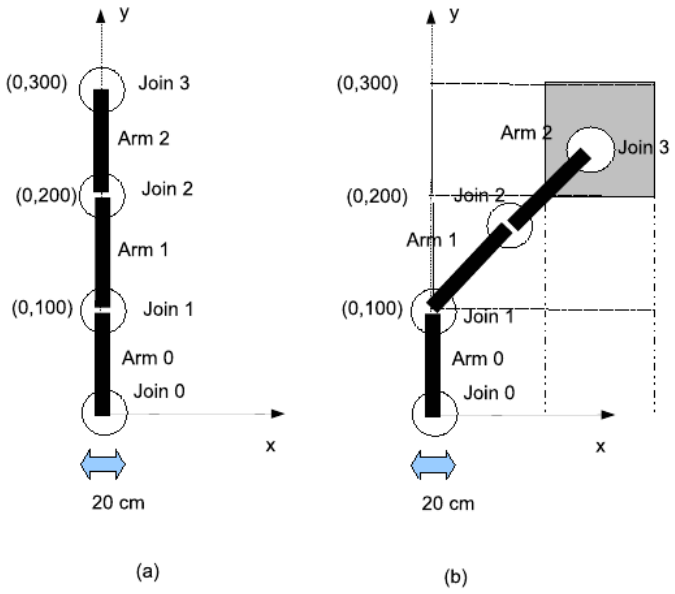

Figure 2: A 2D simplified robot arm.

Input

The test data begins with a positive integer M, where M is the number of test cases. Each test case begins with a positive integer N, which is the number of arms. The last part of a test case is the rectangle area specified by $x_1\ y_1\ x_2\ y_2$ where (x_1, y_1) is the coordinate of bottom left corner and (x_2, y_2) is the coordinate of top-right corner.

Output

Please output the minimum number of moves for each test case. If a test case does not have feasibile solution, please output '`-1`'.

Sample Input

```
3
1
100 200 200 300
3
50 150 150 250
4
150 -50 250 50
```

Sample Output

```
-1
2
3
```