

/*

The game of billiards involves two players knocking 3 balls around on a green baize table.

Well, there is more to it, but for our purposes this is sufficient.

The game consists of several rounds and in each round both players obtain a score, based on how well they played.

Once all the rounds have been played, the total score of each player is determined by adding up the scores in all the rounds

and the player with the higher total score is declared the winner.

The MPSTME Student Sports Committee organises an annual billiards game where the top two players of MPSTME Student Sports Committee play against each other.

The faculty incharge of MPSTME Student Sports Committee decided to add his own twist to the game by changing the rules for determining the winner.

In his version, at the end of each round, the cumulative score for each player is calculated, and the leader and her current lead are found.

Once all the rounds are over the player who had the maximum lead at the end of any round in the game is declared the winner.

The winner of this game is Player 1 as he had the maximum lead (58 at the end of round 1) during the game.

Your task is to help the Manager find the winner and the winning lead. You may assume that the scores will be such that there will always be a single winner. That is, there are no ties.

*/

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int n;
```

```
    cout << "Enter number of rounds: ";
```

```
    cin >> n;
```

```
    int p1 = 0, p2 = 0, max_lead = 0, winner = 1;
```

```
    for (int i = 0; i < n; i++)
```

```
    {
```

```
        int s1, s2;
```

```
        cout << "Enter score of player 1 and player 2: ";
```

```
        cin >> s1 >> s2;
```

```
        p1 += s1;
```

```
        p2 += s2;
```

```
        int lead = p1 - p2;
```

```
        if (lead > max_lead)
```

```
        {
```

```
            max_lead = lead;
```

```
            winner = 1;
```

```
        }
```

```
        if (lead < -max_lead)
```

```
        {
```

```
        max_lead = -lead;
        winner = 2;
    }
}
cout << "The winner is player " << winner << " with a lead of " << max_lead << endl;
return 0;
}
```

```
Enter number of rounds: 3
Enter score of player 1 and player 2: 2
4
Enter score of player 1 and player 2: 5
3
Enter score of player 1 and player 2: 8
2
The winner is player 1 with a lead of 6
```