**LEVEL 2 - PROVE YOURSELF**

**Same place, the next day. You have joined Jeff and Mike in the crisis meeting room of the Kennedy Space Center.**

*“OK, I see you got the general idea. Mike, what do you think of our new recruit so far?”  
“There's still a long way to go.”  
“Oh c'mon Mike, you're always so skeptical!”*

Jeff turns and glares at you with his steel-blue eyes.

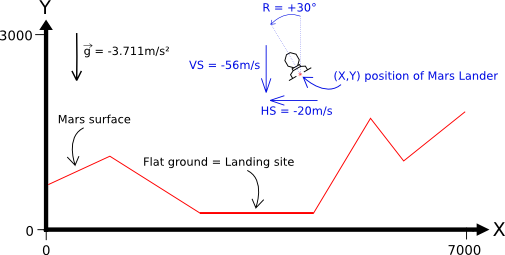
*“But he IS right! This first test was just a warm-up. Now you'll need to deal with more challenging situations. You see, we must be prepared to face anything, the success of the mission depends upon it!”*

A new set of **five more complex tests** awaits you.  
Do not hesitate to **click on "Previous question" to copy your code** and paste it back in the editor of this new level.

**Warning**: this time there is more than one test. So before submitting your final code use the "**Test script**" window on the bottom right hand corner of the screen to switch between tests by changing the value of the "test" variable (1, 2, 3, 4, or 5).

In fact the problems concerns **the landing phase**for “Mars Lander”, the landing ship which contains the Opportunity rover. Mars Lander is guided by a program, and right now the failure rate for landing on the NASA simulator is unacceptable.

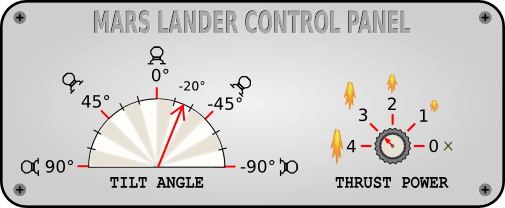
Built as a game, the simulator puts Mars Lander on a limited zone of Mars sky. The zone is 7000m wide and 3000m high. The ship can get into the zone at a variable location and with a variable speed and tilt angle.



Your mission is to write a new artificial intelligence program that will enable Mars Lander to land safely on Mars without crashing. The program will have to go through a series of increasingly complex simulator tests.

**THE PROGRAM:**

**Every second**, depending on the current flight parameters (location, speed, fuel ...), the program must provide the new desired tilt angle and thrust power of Mars Lander.

  
Angle goes from -90° to 90°. Thrust power goes from 0 to 4.

The game simulates **a free fall** without atmosphere. Gravity on Mars is 3.711 m/s². For a **thrust power of X**, a push force equivalent to **X m/s²** is generated and **X liters of fuel** are consumed. As such, a thrust power of 4 in an almost vertical position is needed to compensate for the gravity on Mars.  
  
For a landing to be successful, the ship must:

* land on flat ground
* land in a vertical position (tilt angle = 0°)
* vertical speed must be limited ( ≤ 40m/s in absolute value)
* horizontal speed must be limited ( ≤ 20m/s in absolute value)

For each test, there is a **unique area of flat ground** on the surface of Mars which is at least 1000 meters wide.  
  
The program must first read the initialization data from standard input. Then, within an infinite loop, the program must read the data from the standard input related to Mars Lander's current state and provide to the standard output the instructions to move Mars Lander.

**INITILIZATION INPUT:**

**Line 1:** the number **N** of points used to draw the surface of Mars.

**N following lines:** a couple of integers **X Y** providing the coordinates of a ground point. By linking all the points together in a sequential fashion, you form the surface of Mars which is composed of several segments. For the first point, X = 0 and for the last point, X = 6999

**INPUT FOR ONE GAME TURN:**

A **single line** with 7 integers: **X Y HS VS F R P**

* (X, Y) are the coordinates of Mars Lander (in meters).
* HS et VS are the horizontal and vertical speed of Mars Lander (in m/s). These can be negative depending on the direction of Mars Lander.
* F is the remaining quantity of fuel in liters. When there is no more fuel, the power of thrusters falls to zero.
* R is the angle of rotation of Mars Lander expressed in degrees.
* P is the thrust power of the landing ship.

**OUTPUT FOR ONE GAME TURN:**

A **single line** with 2 integers: **R P**

* R is the desired rotation angle for Mars Lander. Please note that for each turn the actual value of the angle is limited to the value of the previous turn +/- 15°.
* P is the desired thrust power. 0 = off. 4 = maximum power. Please note that for each turn the value of the actual power is limited to the value of the previous turn +/- 1.

**CONSTRAINTS:**

2 ≤ N < 30  
0 ≤ X < 7000  
0 ≤ Y < 3000  
-500 < HS, VS < 500  
0 ≤ F ≤ 2000  
-90 ≤ R ≤ 90  
0 ≤ P ≤ 4

Answer time per turn ≤ 100ms

**EXAMPLE:**

Mars Lander starts with a vertical speed of zero, a horizontal speed of -50m/s (going left) and is tilted fully to the left (90°).

|  |  |
| --- | --- |
| **Initialization input** *(out of the infinite loop)* | ***No output expected*** |
| |  |  | | --- | --- | | 6 | *(N) Surface made of 6 points* | | 0 1500 | *(X Y)* | | 1000 2000 | *(X Y)* | | 2000 500 | *(X Y) Start of flat ground* | | 3500 500 | *(X Y) End of flat ground* | | 5000 1500 | *(X Y)* | | 6999 1000 | *(X Y)* | |  |
| **Input for turn 1** | **Output for turn 1** |
| |  |  | | --- | --- | | 5000 2500 -50 0 1000 90 0 | *(X Y HS VS F R P)* | | -45 4 *(R P) Requested rotation to the right, maximum thrust power* |
| **Input for turn 2** | **Output for turn 2** |
| |  |  | | --- | --- | | 4950 2498 -51 -3 999 75 1 | *(X Y HS VS F R P) Tilt angle changed only by 15° and thrust power only by 1* | | -45 4 *(R P) Same request as previous turn* |
| **Input for turn 3** | **Output for turn 3** |
| |  |  | | --- | --- | | 4898 2493 -53 -6 997 60 2 | *(X Y HS VS F R P)* | | -45 4 *(R P) Same request as previous turn* |
| * and so on until Mars Lander crashes or lands... | |