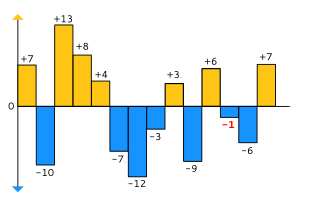
Dans cet exercice, vous devez analyser un relevé de températures pour trouver quelle température se rapproche le plus de zéro.



Exemple de températures

Ici, -1 est le plus proche de 0.

**Règles**

Écrivez un programme qui affiche la température la plus proche de 0 parmi les données d'entrée. **Si deux nombres sont aussi proches de zéro, alors l'entier positif sera considéré comme étant le plus proche de zéro** (par exemple, si les températures sont -5 et 5, alors afficher 5).

<?php

/\*\*

\* Auto-generated code below aims at helping you parse

\* the standard input according to the problem statement.

\*\*/

**Solution**

fgets(STDIN);

$list = explode(' ',fgets(STDIN));

for ($temp = 0; $temp <= 5526; $temp++)

{

if (in\_array($temp, $list)) exit ("$temp");

if (in\_array(-$temp, $list)) exit ("-$temp");

}

**Ma solution**

$temp = 5527;

// $n: the number of temperatures to analyse

fscanf(STDIN, "%d", $n);

$inputs = explode(" ", fgets(STDIN));

for ($i = 0; $i < $n; $i++)

{

$t = intval($inputs[$i]); // a temperature expressed as an integer ranging from -273 to 5526

if ($temp > abs($t)) {

$temp = abs($t);

if ($t < 0) {

$result = $temp \* -1;

} else {

$result = $temp;

}

} elseif ($temp == abs($t)) {

if ($result < $t) {

$result = $temp;

}

}

}

if ($temp == 5527) {

$result = 0;

}

// Write an action using echo(). DON'T FORGET THE TRAILING \n

// To debug (equivalent to var\_dump): error\_log(var\_export($var, true));

echo("$result\n");

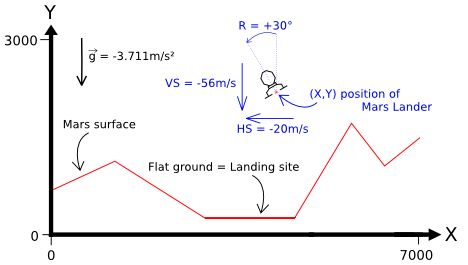
?>

**Objectif**

L'objectif de votre programme est de faire atterrir, sans crash, la capsule "Mars Lander" qui contient le rover Opportunity. La capsule “Mars Lander” permettant de débarquer le rover est pilotée par un programme qui échoue trop souvent dans le simulateur de la NASA.  
  
**Notez que ce problème peut sembler difficile, mais en réalité il est simple à résoudre.** Ce puzzle constitue le premier des trois niveaux, par conséquent, certains contrôles sont présentés mais ne sont pas nécessaires pour résoudre ce premier niveau.

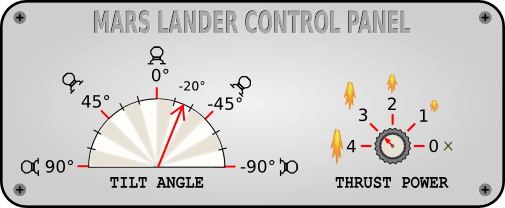
**Règles**

Sous forme de jeu, le simulateur place Mars Lander dans une zone du ciel de Mars.



La zone fait 7000m de large et 3000m de haut.  
  
**Pour ce niveau,** Mars Lander se situe au dessus de la zone d’atterrissage, en position verticale, avec aucune vitesse initiale.  
  
Il existe une **unique zone d'atterrissage plane** sur la surface de Mars et elle mesure au moins 1000 mètres de large.

**Toutes les secondes**, en fonction des paramètres d’entrée (position, vitesse, fuel, etc.), le programme doit fournir le nouvel angle de rotation souhaité ainsi que la nouvelle puissance des fusées de Mars Lander:



Angle de -90° à 90°. Puissance des fusées de 0 à 4.  
  
**Pour ce niveau,** vous n'avez besoin de contrôler que la puissance des fusées : l'angle doit rester à 0.

Le jeu modélise **une chute libre** sans atmosphère. La gravité sur Mars est de 3,711 m/s². Pour une **puissance des fusées de X**, on génère une poussée équivalente à **X m/s²** et on consomme **X litres de fuel**. Il faut donc une poussée de 4 quasi verticale pour compenser la gravité de Mars.  
  
Pour qu’un atterrissage soit réussi, la capsule doit :

* atterrir sur un sol plat
* atterrir dans une position verticale (angle = 0°)
* la vitesse verticale doit être limitée ( ≤ 40 m/s en valeur absolue)
* la vitesse horizontale doit être limitée ( ≤ 20 m/s en valeur absolue)

**Souvenez-vous que ce puzzle a été simplifié, ainsi :**

* la zone d'atterrissage est juste en dessous du robot. Vous pouvez donc ignorer la rotation et toujours indiquer 0 en angle de rotation.
* vous n'avez pas besoin de tenir compte des coordonnées de la surface.
* il vous suffit que votre vitesse d'atterrissage soit entre 0 et 40m/s.
* lorsque la capsule descend vers le sol, la vitesse verticale est négative. Lorsque la capsule s'élève dans les airs, la vitesse verticale est positive.

**Ma solution**