

Coding Game Descent

Lien :

Coding Games and Programming Challenges to Code Better

CodinGame is a challenge-based training platform for programmers where you can play with the hottest programming topics. Solve games, code AI bots, learn from your peers, have fun.

 <https://www.codinggame.com/ide/puzzle/the-descent>

Play **coding games**.
Have fun. Build skills.



Level up your coding with games, puzzles, and challenges.

Consignes :

🎯 Objectif

Écrivez le programme permettant de détruire les montagnes pour pouvoir atterrir. Pour cela, tirez sur la montagne la plus haute.

✓ Règles

Au début de chaque tour de jeu, vous recevez en entrée la hauteur de chaque montagne de gauche à droite.
Avant la fin du tour de jeu, vous devez indiquer la montagne la plus haute pour tirer dessus.

Tirer sur une montagne ne fera qu'en détruire une partie. Votre vaisseau descend à chaque passe.



Conditions de victoire

Vous gagnez si vous détruisez la montagne la plus haute à chaque tour



Conditions de défaite

- Votre vaisseau percute une montagne
- Vous fournissez une sortie invalide ou votre programme ne réponds pas à temps

■ Entrées du jeu

Le programme doit lire depuis l'entrée standard la hauteur des montagnes puis fournir sur la sortie standard l'indice de la montagne à détruire.

Entrées pour un tour de jeu

8 lignes : un entier `mountainH` par ligne. Il représente la hauteur d'une montagne. Les hauteurs des montagnes sont données dans l'ordre de leur index (allant de 0 à 7).

Sortie pour un tour de jeu

Une unique ligne contenant le numéro de la montagne sur laquelle tirer.

Contraintes

$0 \leq \text{mountainH} \leq 9$

Temps de réponse pour un tour $\leq 100\text{ms}$

Début de code :

```
import sys
import math

# The while loop represents the game.
# Each iteration represents a turn of the game
# where you are given inputs (the heights of the mountains)
# and where you have to print an output (the index of the mountain to fire on)
# The inputs you are given are automatically updated according to your last actions.

# game loop
while True:
    for i in range(8):
        mountain_h = int(input()) # represents the height of one mountain.

    # Write an action using print
    # To debug: print("Debug messages...", file=sys.stderr, flush=True)

    # The index of the mountain to fire on.
    print("4")
```

Dans ce début de code, nous avons une boucle du jeu qui dit :



```
while True :
    for i in range(8):
        mountain_h = int(input()) # represents the height of one mountain.
```

Si vrai, pour chaque valeur de `i` de 0 à 7, demandez la hauteur de la montagne à chaque itération et enregistrez cette hauteur dans `mountain_h`.



```
print("4")
```

Afficher "4" qui veut dire dans le concept du jeu, tire sur la montagne 4.

En exécutant le code voici ce que nous avons en sortie de console :

```
Sortie console Informations de jeu, ...
Informations :
Let's destroy those mountains to secure our landing...

Height of mountain 0 : 9
Height of mountain 1 : 8
Height of mountain 2 : 7
Height of mountain 3 : 6
Height of mountain 4 : 5
Height of mountain 5 : 4
Height of mountain 6 : 3
Height of mountain 7 : 2

Sortie standard :
> 4
Informations :
Your spaceship targeted mountain 4, your altitude is now 9

Height of mountain 0 : 9
Height of mountain 1 : 8
Height of mountain 2 : 7
Height of mountain 3 : 6
Height of mountain 4 : 0
Height of mountain 5 : 4
Height of mountain 6 : 3
Height of mountain 7 : 2

Sortie standard :
> 4
Informations :
You crashed your spaceship on mountain 0

Height of mountain 0 : 9
Height of mountain 1 : 8
Height of mountain 2 : 7
Height of mountain 3 : 6
Height of mountain 4 : 0
Height of mountain 5 : 4
Height of mountain 6 : 3
Height of mountain 7 : 2
```

Partie 1

Partie 2

Partie 3

Partie 1 :

On peut constater qu'il y'a bien 8 montagne (de 0 à 7).

Partie 2 :

Nous débutons le jeu à partir de l'altitude de 10.

La sortie est print(4) qui veut dire tirer sur la montagne 4.

La montagne 4 représente une altitude de 5 (9 est l'altitude le plus élevé pour une montagne)

Partie 3 :

Notre avion s'est crashé sur la montagne 0 après avoir éliminer la montagne 4 (représentant une boucle)

Pourquoi ? Il s'est crashé car après une boucle il redescend d'un altitude de -1 ce qui veut dire après avoir éliminer la montagne 4 il passe de l'altitude 10 à 9.

La montagne 0 représentant une hauteur d'altitude de 9 fait crasher l'avion.

Voici ma solution et ma méthodologie pour ce test

Mon code en python :

```
import sys
import math

# The while loop represents the game.
# Each iteration represents a turn of the game
# where you are given inputs (the heights of the mountains)
# and where you have to print an output (the index of the mountain to fire on)
# The inputs you are given are automatically updated according to your last actions.

# Boucle du jeu
while True:
    numéro_plus_haute_montagne = 0 # Variable de la première montagne
    hauteur_max = -1 # Variable pour se souvenir de la hauteur de la montagne la plus haute

    for i in range(8): # Pour chaque montagne (on en a 8 au total)
        mountain_h = int(input()) # représente la hauteur d'une montagne.

        # On vérifie si c'est le numéro plus haute montagne qu'on voit ou si elle est plus haute que les autres
        if i == 0 or hauteur_max < mountain_h:
            numéro_plus_haute_montagne = i # On se souvient du numéro de cette montagne
            hauteur_max = mountain_h # On se souvient de sa hauteur

    # À ce stade, on sait quelle montagne est la plus haute. On va dire à l'avion laquelle il faut viser.

    # Le numéro de la montagne à viser.
    print(numéro_plus_haute_montagne) # On dit à l'avion le numéro de la montagne à tirer
```

La méthodologie :

Variable créer :

Ces deux variables, `numéro_plus_haute_montagne` et `hauteur_max`, sont utilisées pour garder en mémoire l'information sur la montagne la plus haute que le programme a vue jusqu'à présent. Voici comment cela

fonctionne :

1. `numéro_plus_haute_montagne` : Cette variable garde en mémoire le numéro de la montagne qui est actuellement considérée comme la plus haute. Au début, on l'initialise à zéro, car on n'a encore vu aucune montagne. Puis, à mesure que l'on observe les montagnes, si on en voit une qui est plus haute que celle que l'on considérait comme la plus haute, on met à jour cette variable avec le numéro de la nouvelle montagne.
2. `hauteur_max` : Cette variable conserve la hauteur de la montagne la plus haute que l'on a vue jusqu'à présent. On l'initialise à -1, car la hauteur d'une montagne sera toujours positive. Cela signifie qu'à partir du moment où on voit la première montagne, sa hauteur sera toujours plus grande que -1. Puis, à chaque fois qu'on voit une nouvelle montagne, on compare sa hauteur avec la hauteur maximale enregistrée. Si elle est plus grande, on met à jour `hauteur_max` avec la nouvelle hauteur.

Conditions :

1. **Première Montagne** : La première fois que tu vois une montagne (`i == 0`), tu n'as pas vu d'autres montagnes encore. Donc, tu te souviens automatiquement du numéro de cette montagne (`numéro_plus_haute_montagne = i`) et de sa hauteur (`hauteur_max = mountain_h`).
2. **Comparaisons Ultérieures** : À partir de la deuxième montagne que tu vois (lorsque `i` n'est plus égal à zéro), tu compares la hauteur de cette montagne (`mountain_h`) avec la hauteur de la montagne la plus haute que tu as vue jusqu'à présent (`hauteur_max`). Si la montagne actuelle est plus haute, alors tu mets à jour le numéro de la montagne la plus haute (`numéro_plus_haute_montagne = i`) et tu mets à jour la hauteur maximale (`hauteur_max = mountain_h`).

Cela signifie que tu gardes en mémoire la montagne la plus haute que tu as vue jusqu'à présent. À chaque nouvelle montagne que tu vois, tu vérifies si elle est plus haute que celle que tu avais en mémoire. Si c'est le cas, alors tu mets à jour tes souvenirs avec les détails de cette nouvelle montagne plus haute.

En fin de compte, une fois que tu as regardé toutes les montagnes, tu te souviens de celle qui est la plus haute, et c'est celle-là que tu vas viser.

C'est un peu comme si tu prenais une photo mentale de chaque montagne que tu vois. Si tu vois une montagne plus haute que celles que tu as vues auparavant, tu jettes ta vieille photo et prends une nouvelle photo de la montagne plus haute.

À la fin, tu as une photo de la montagne la plus haute que tu as vue, et c'est celle-là que tu choisis.