Jeux d’essais

# But du laboratoire

* Utiliser des jeux d’essais
* Utiliser de l’outil de débogage
* Créer des fonctions
* Lire le code d’un autre programmeur

# Tâche

Tester est une partie importante et souvent négligée de la programmation. Dans cet exercice, vous devrez tester du code créé par un autre programmeur et trouver les problèmes. Vous devez remplir le jeu d’essais correspondant avant de tester son application. Vous devez ensuite trouver les erreurs, les réparer et écrire vos découvertes dans les encadrés ci-dessous.

## Exemple : jeux d’essais 1

Créez une fonction qui permet d’entrer un ou deux nombres entiers et qui additionne leurs valeurs. Si un seul nombre est entré la valeur du deuxième est de 5 par défaut.

### Variables à surveiller

|  |  |  |
| --- | --- | --- |
| **Nom de la variable** | **Description** | **Type et valeur par défaut** |
| nombre1 | Premier nombre à additionner | Integer |
| nombre2 | Second nombre à additionner | Integer  Défaut: 5 |

### Jeu d’essais

|  |  |  |  |
| --- | --- | --- | --- |
| **nombre1** | **nombre2** | **Attendu** | **Obtenu** |
| 3 |  | 8 | 8 |
| 3 | 9 | 12 | 12 |
| -3 | 9 | 6 | 12 |
| 3 | -9 | -6 | 6 |

Erreur 1 :

Type d’erreur : Erreur de logique.

Mauvais code : Ligne 9.

resultat = abs(nombre1) + abs(nombre2)

Bon code :

resultat = nombre1 + nombre2

Explication :

Convertir les nombres entiers dans leur valeur absolue retourne toujours un nombre positif.

## Jeux d’essais 2

Créer une fonction qui compare la valeur d’un nombre entré avec celui du précédent. La valeur initiale pour la comparaison est 0. Vous devez afficher si le nombre est plus petit, plus grand ou égal au nombre précédent. Mettez la fonction dans une boucle infinie jusqu’à ce que l’entrée ne soit pas un nombre réel.

### Variables à surveiller

|  |  |  |
| --- | --- | --- |
| **Nom de la variable** | **Description** | **Type et valeur par défaut** |
| Nombre 1 | Premier nombre à comparer | Float 0 |
| Nombre 2 | Deuxième nombre à comparer | float |

### Jeu d’essais

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre 1** | **Nombre 2** | **Attendu** | **Obtenu** |
| 0 | 235 | Nombre2 > Nomber1 | Nombre2 > Nomber1 |
| 235 | 12 | Nombre2 <  Nomber1 | Nombre2 <  Nomber1 |
| 12 | 12 | Nombre2 = Nomber1 | Nombre2 <  Nomber1 |
| 12 | -1 | Nombre2 > Nomber1 | Ne compare pas |

Erreur 1 :

Type d’erreur : Erreur de Logique

Mauvais code : if nombre2 < nombre1:  
 return f"{nombre2} est plus petit que {nombre1}."  
else:  
 return f"{nombre2} est plus grand que {nombre1}."

Bon code : if nombre2 < nombre1:  
 return f"{nombre2} est plus petit que {nombre1}."  
elif nombre2 == nombre1:  
 return f"{nombre2} est égale à {nombre1}."

Else :

return f"{nombre2} est plus grand que {nombre1}."

Explication : Ceci permet les 2 chiffres comparés à etre égales. S’il n’est pas réparer, le code sortira de la boucle.