



Épreuve de l'Eau

Consignes

- Seul les consignes de ce document sont vérifiées : n'écoutez pas les rumeurs.
- Tous les sujets sont susceptibles d'évoluer.
- Testez votre programme en étant sans pitié avant de demander une correction, vos correcteurs seront sans pitié aussi.
- Respectez la norme de votre langage.
- N'utilisez pas des fonctions qui résolvent le problème, vous êtes là pour apprendre.
- Vous trouverez toutes les connaissances supplémentaires dont vous avez besoin sur Google.
- Faites preuve de réflexion.

Combinaisons de 3 chiffres

Créez un programme qui affiche toutes les différentes combinaisons possibles de trois chiffres dans l'ordre croissant, dans l'ordre croissant. La répétition est volontaire.

Exemples d'utilisation :

```
$> python exo.py  
012, 013, 014, 015, 016, 017, 018, 019, 023, 024, ... , 789  
$>
```

987 n'est pas là parce que 789 est présent.

020 n'est pas là parce que 0 apparaît deux fois.

021 n'est pas là parce que 012 est présent.

000 n'est pas là parce que cette combinaison ne comporte pas exclusivement des chiffres différents les uns des autres.

Combinaisons de 2 nombres

Créez un programme qui affiche toutes les différentes combinaisons de deux nombre entre 00 et 99 dans l'ordre croissant.

Exemples d'utilisation :

```
$> python exo.py  
00 01, 00 02, 00 03, 00 04, ... , 00 99, 01 02, ... , 97 99, 98 99  
$>
```

Paramètres à l'envers

Créez un programme qui affiche ses arguments reçus à l'envers.

Exemples d'utilisation :

```
$> python exo.py "Suis" "Je" "Drôle"  
Drôle  
Je  
Suis
```

```
$> python exo.py ha ho  
ho  
ha
```

```
$> python exo.py "Bonjour 36"  
Bonjour 36
```

Afficher error et quitter le programme en cas de problèmes d'arguments.

Suite de Fibonacci

Créez un programme qui affiche le N-ème élément de la célèbre suite de Fibonacci. (0, 1, 1, 2) étant le début de la suite et le premier élément étant à l'index 0.

Exemples d'utilisation :

```
$> python exo.py 3  
2  
$>
```

Afficher -1 si le paramètre est négatif ou mauvais.

Prochain nombre premier

Créez un programme qui affiche le premier nombre premier supérieur au nombre donné en argument.

Exemples d'utilisation :

```
$> python exo.py 14  
17  
$>
```

Afficher -1 si le paramètre est négatif ou mauvais.

String dans string

Créez un programme qui détermine si une chaîne de caractère se trouve dans une autre.

Exemples d'utilisation :

```
$> python exo.py bonjour jour  
true
```

```
$> python exo.py bonjour joue  
false
```

```
$> python exo.py 42  
error
```

Afficher error et quitter le programme en cas de problèmes d'arguments.

Majuscule sur deux

Créez un programme qui met en majuscule une lettre sur deux d'une chaîne de caractères. Seule les lettres (A-Z, a-z) sont prises en compte.

Exemples d'utilisation :

```
$> python exo.py "Hello world !"  
HeLlO wOrLd !
```

```
$> python exo.py 42  
error
```

Afficher error et quitter le programme en cas de problèmes d'arguments.

Majuscule

Créez un programme qui met en majuscule la première lettre de chaque mot d'une chaîne de caractères. Les autres lettres devront être en minuscules. Les mots ne sont délimités que par un espace, une tabulation ou un retour à la ligne.

Exemples d'utilisation :

```
$> python exo.py "bonjour mathilde, comment vas-tu ?!"  
Bonjour Mathilde, Comment Vas-tu ?!
```

```
$> python exo.py 42  
error
```

Afficher error et quitter le programme en cas de problèmes d'arguments.

Chiffres only

Créez un programme qui détermine si une chaîne de caractères ne contient que des chiffres.

Exemples d'utilisation :

```
$> python exo.py "4445353"  
true
```

```
$> python exo.py 42  
true
```

```
$> python exo.py "Bonjour 36"  
false
```

Afficher error et quitter le programme en cas de problèmes d'arguments.

Entre min et max

Créez un programme qui affiche toutes les valeurs comprises entre deux nombres dans l'ordre croissant. Min inclus, max exclus.

Exemples d'utilisation :

```
$> python exo.py 20 25  
20 21 22 23 24
```

```
$> python exo.py 25 20  
20 21 22 23 24
```

```
$> python exo.py hello  
error
```

Afficher error et quitter le programme en cas de problèmes d'arguments.

Index wanted

Créez un programme qui affiche le premier index d'un élément recherché dans un tableau. Le tableau est constitué de tous les arguments sauf le dernier. L'élément recherché est le dernier argument. Afficher -1 si l'élément n'est pas trouvé.

Exemples d'utilisation :

```
$> python exo.py Ceci est le monde qui contient Charlie un homme sympa Charlie  
6
```

```
$> python exo.py test test test  
0
```

```
$> python exo.py test boom  
-1
```

Afficher error et quitter le programme en cas de problèmes d'arguments.

Différence minimum absolue

Créez un programme qui affiche la différence minimum absolue entre deux éléments d'un tableau constitué uniquement de nombres. Nombres négatifs acceptés.

Exemples d'utilisation :

```
$> python exo.py 5 1 19 21  
2
```

```
$> python exo.py 20 5 1 19 21  
1
```

```
$> python exo.py -8 -6 4  
2
```

Afficher error et quitter le programme en cas de problèmes d'arguments.

Tri à bulle

Créez un programme qui trie une liste de nombres. Votre programme devra implémenter l'algorithme du tri à bulle.

Vous utiliserez une fonction de cette forme (selon votre langage) :

```
my_bubble_sort(array) {  
    # votre algorithme  
    return (new_array)  
}
```

Exemples d'utilisation :

```
$> python exo.py 6 5 4 3 2 1  
1 2 3 4 5 6
```

```
$> python exo.py test test test  
error
```

Afficher error et quitter le programme en cas de problèmes d'arguments.

Wikipedia vous présentera une belle description de cet algorithme de tri.

Tri par sélection

Créez un programme qui trie une liste de nombres. Votre programme devra implémenter l'algorithme du tri par sélection.

Vous utiliserez une fonction de cette forme (selon votre langage) :

```
my_select_sort(array) {  
    # votre algorithme  
    return (new_array)  
}
```

Exemples d'utilisation :

```
$> python exo.py 6 5 4 3 2 1  
1 2 3 4 5 6
```

```
$> python exo.py test test test  
error
```

Afficher error et quitter le programme en cas de problèmes d'arguments.

Wikipedia vous présentera une belle description de cet algorithme de tri.

Par ordre Ascii

Créez un programme qui trie les éléments donnés en argument par ordre ASCII.

Exemples d'utilisation :

```
$> python exo.py Alfred Momo Gilbert  
Alfred Gilbert Momo
```

```
$> python exo.py A Z E R T Y  
A E R T Y Z
```

Afficher error et quitter le programme en cas de problèmes d'arguments.

Eau : ok

Créez un programme qui célèbre votre victoire.

Exemples d'utilisation :

```
$> ruby exo.rb
```

```
J'ai terminé l'Épreuve de l'Eau et c'était [].
```

```
$>
```

Note : [] est à remplacer par un adjectif de votre choix (facile ?)