~ Contrôle ~

France ioi – validation

Nom :	Prénom :	Classe :
Remarque :	Note :	

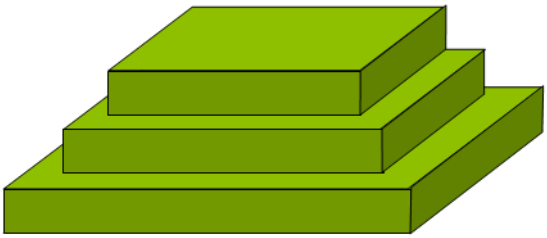
Réaliser sur feuille 3 exercices parmis les 5 proposés. La solution à chaque enigme doit être justifiée.
Calculatrice, téléphone, ordinateur interdits. Durée : 55 minutes.

1 Lecture de l'entrée

Les habitants d'Algoréa aiment bien ériger des statues et les poser sur des socles majestueux. Selon les dimensions de la statue, le socle doit être plus ou moins haut et offrir une surface plus ou moins grande pour y poser la statue. Cependant, les constructeurs de statues ont souvent du mal à estimer la quantité de béton nécessaire à la construction de chaque socle. Vous souhaitez écrire un programme pour les aider.

Ce que doit faire votre programme :

Voici un exemple de socle :



Un socle est ainsi constitué d'étages, chaque étage ayant une hauteur égale à une unité et une base carrée. Le côté des carrés diminue de une unité à chaque étage.

Votre programme doit lire deux entiers, représentant respectivement la largeur du socle au niveau du sol et la largeur du socle au niveau de la face supérieure du socle. Il doit ensuite calculer et afficher le volume du socle.

Exemple

entrée :

```
7
3
```

sortie :

```
135
```

Commentaires

Le volume du premier étage est 7 × 7 = 49, le volume du second est 6 × 6 = 36, etc. et le volume du dernier étage est 3 × 3 = 9. Le volume total est donc : 7 × 7 + 6 × 6 + 5 × 5 + 4 × 4 + 3 × 3 = 135.

2 Tests et conditions

Vous avez passé la nuit dans une auberge. Au petit matin, un championnat de tir à la corde y est organisé. Vous ne souhaitez pas participer, mais l'aubergiste insiste pour que vous soyez impliqué dans l'événement. Vous décidez alors de vous engager dans les paris qui se font sur les deux équipes qui concourent.

Le championnat oppose deux équipes, contenant chacune autant de joueurs. Pour donner de l'allure et pimenter les paris, au début du championnat, tous les joueurs sont présentés, avec leur poids, avant d'aller tenir leur côté de la corde. Il est d'abord présenté un membre de la première équipe, puis de la deuxième, puis de la première, puis de la deuxième etc. jusqu'à ce que tous les joueurs soient passés. Afin de vous faire un premier pronostic, vous calculez le poids total de chaque équipe, avec votre robot.

Ce que doit faire votre programme :

Votre programme devra lire un premier entier : le nombre de membres *nbMembres* qui constituent une équipe. Ensuite, il devra lire les poids (en kilogrammes), au total *nbMembres* × 2, sachant que le premier poids est celui d'un joueur de la 1re équipe, le deuxième poids celui d'un joueur de la 2e équipe, le troisième la 1re équipe, le quatrième la 2e équipe, etc.

Après avoir calculé le poids total de chaque équipe, vous devrez afficher le texte « L'équipe *X* a un avantage » (en remplaçant *X* par la valeur 1 ou 2), en considérant qu'une équipe est avantagée si elle a un poids total supérieur à celui de l'autre.

Vous afficherez ensuite le texte « Poids total pour l'équipe 1 : » suivi du poids de l'équipe 1, puis « Poids total pour l'équipe 2 : » suivi du poids de l'équipe 2 (voir l'exemple ci-dessous).

On vous garantit que les deux équipes n'auront pas le même poids total.

Exemple

entrée :

```
3
40
80
50
50
60
10
```

sortie :

```
L'équipe 1 a un avantage
Poids total pour l'équipe 1 : 150
Poids total pour l'équipe 2 : 140
```

Commentaires

Chaque équipe est composée de trois joueurs. Ceux de la première pèsent 40, 50 et 60 kg, tandis que ceux de la seconde font 80, 50 et 10 kg. Cela fait 150 kg opposés à 140 kg.

3 Structures avancées

Alors que vous traversez une forêt vous ne pouvez vous empêcher d'admirer la végétation autour de vous et notamment les nombreuses espèces d'arbres. Malgré votre intérêt, vous êtes très mauvais botaniste et avez beaucoup de mal à identifier les différents arbres. Une personne que vous croisez vous donne quelques indications et vous décidez d'écrire un programme qui vous donnera le nom de l'arbre en fonction de ses caractéristiques.

Ce que doit faire votre programme :

Il existe 4 types d'arbres :

- le "Tinuviel" fait moins de 5 mètres de haut et ses feuilles sont composées de plus de 8 folioles
- le "Calaelen" fait plus de 10 mètres de haut et ses feuilles sont composées de plus 10 folioles
- le "Falarion" fait moins de 8 mètres de haut et ses feuilles sont composées de moins de 5 folioles
- le "Dorthonion" fait plus de 12 mètres de haut et ses feuilles sont composées de moins de 7 folioles

Votre programme lira deux entiers, la hauteur et le nombre de folioles de l'arbre, et affichera le nom de l'arbre correspondant.

Toutes les inégalités sont à prendre au sens large, c'est-à-dire que "moins" signifie "moins ou égal" ou et "plus" signifie "plus ou égal".

Exemples

Exemple 1

entrée :

```
12
12
```

sortie :

```
Calaelen
```

Exemple 2

entrée :

```
4
9
```

sortie :

```
Tinuviel
```

4 Conditions avancées, opérateurs booléens

Un personnage important de la cité n'est pas rentré chez lui hier soir et tout le monde est à sa recherche. Or tout habitant de la ville a un numéro unique qui lui est associé et doit signer une sorte de registre quand il sort de la ville. Vous souhaitez savoir si le registre a été signé, auquel cas il faudra étendre les recherches à l'extérieur de la ville.

Ce que doit faire votre programme :

On vous donne un entier, le numéro d'une personne recherchée, puis un entier *tailleListe*, et enfin *tailleListe* entiers parmi lesquels vous devez chercher le numéro de la personne. Si le numéro est présent dans la liste (il peut l'être plusieurs fois) vous devez afficher le texte "Sorti de la ville" sinon "Encore dans la ville".

Exemple

entrée :

```
42
5
1
7
172
2
41
```

sortie :

```
Encore dans la ville
```

Les booléens

En Python le programme suivant :

```
if prix < 10:
    print("Pas cher")
```

peut aussi s'écrire

```
estPasCher = (prix < 10)
if estPasCher:
    print("Pas cher")
```

On a donc stocké dans la variable `estPasCher` la valeur de la comparaison `prix < 10` pour la réutiliser plus tard dans un `if`.

La variable `estPasCher` est appelée une *variable booléenne* ou un *booléen* car elle ne peut être que vraie ou fausse, ce qui correspond en Python aux valeurs `True` (pour vrai) et `False` (pour faux).

Ainsi en Python, `(3 < 10)` vaut `True` et `(11 == 7)` vaut `False` . `True` et `False` sont donc des constantes du langage, au même titre que 0, 1 ou encore 42.

On peut donc affecter directement la valeur `True` ou `False` à une variable :

```
toujoursVraie = True
if toujoursVraie:
    print("La variable toujoursVraie vaut 'True'")
```

On peut également utiliser les opérateurs `and` et `or` comme le montre l'exemple suivant :

```
estSenior = (age >= 60)
estJeune = (age <= 25) and (age >= 12)
reductionPossible = (estSenior or estJeune)
if reductionPossible:
    print("Réduction!")
```