

Listes

Exercice 1 : **Minimum de liste**

On considère qu'une liste de nombres, notée L , est déjà en mémoire.

Proposer un programme qui détermine l'indice du minimum de L , puis déplace ce minimum en début de liste.

Exercice 2 : **Produit scalaire**

Un vecteur de l'espace sont représenté en machine sous la forme d'une liste de 3 nombres flottants.

Spécifier et écrire un programme qui calcule et affiche le produit scalaire de 2 vecteurs U et V déjà en mémoire.

Par exemple, vous pourrez prendre tout d'abord $u=[1.0, 2.0, 3.0]$ et $v=[4.0, 5.0, 6.0]$. Puis, pour tester votre programme sur d'autres vecteurs, vous changerez les valeurs de u et v directement dans votre programme.

Faire calculer également le cosinus de l'angle entre U et V , et afficher-le.

Mémo, produit scalaire : $\vec{U} \cdot \vec{V} = U_x \times V_x + U_y \times V_y + U_z \times V_z = \|\vec{U}\| \cdot \|\vec{V}\| \cos(\vec{U}, \vec{V})$

On considère que la liste $L=[10, 20, 70, 770, 5620]$ est en mémoire.

Exercice 3 : **Boucles à connaître : application aux listes**

- Q1.** Ecrire un programme qui calcule la somme de tous les éléments de L .
- Q2.** Ecrire un programme qui calcule le produit de tous les éléments de L .
- Q3.** Conjonction : Ecrire un programme qui affiche `True` si tous les nombres d'une liste sont divisible par 10.
- Q4.** Disjonction : Ecrire un programme qui affiche `False` si aucun des nombres d'une liste n'est divisible par 3.
- Q5.** Test : « `x in L` » : en utilisant une boucle `for` ou `while`, proposer un programme qui réalise le ce test. Dans le pire des cas, combien de comparaisons sont nécessaires ?

Chaînes

Exercice 4 : **Programme mystère**

On considère le programme suivant :

```

1 a=478
2 b=25
3 c=str(a)
4 d=str(b)
5 s1=a+b
6 s2=b+c
7 s3=c+d
8 z1=a[2]
9 z2=b[2]
10 z3=c[2]
11 z4=d[2]
```

Ce programme comporte une ou plusieurs erreurs. Quelles lignes entraînent des erreurs d'interprétation. Et pourquoi ?

Exercice 5 : **Damier**

Un damier de dimension 4×4 peut se représenter sous la forme ci-contre :

O	X	O	X
X	O	X	O
O	X	O	X
X	O	X	O

Votre programme doit afficher un damier de taille 40×40. Assurez-vous bien que la case tout en haut à gauche contienne un « O », comme c'est le cas dans le damier ci-dessus.

Conversions

Exercice 6 : **Conversion d'un entier positif en unaire**

La convention de codage « unaire » n'a qu'un chiffre « 1 ». Le nombre d'occurrence du chiffre correspond à la valeur codée. Cela revient à compter sur ses doigts.

- 3 sera codé : 111
- 0 sera codé sans chiffre

Ecrire un programme qui prenne un entier positif indiqué par l'utilisateur et affiche la chaîne de caractères correspondant à son code unaire.

Exercice 7 : **Conversion d'un entier positif en binaire**

Ecrire un programme qui prenne un entier positif indiqué par l'utilisateur et affiche la chaîne de caractères correspondant à son code binaire. (conformément à la convention vue en cours)

Exercice 8 : **Conversion d'un entier relatif en binaire**

Ecrire un programme qui prenne un entier relatif compris entre -128 et 127 inclus indiqué par l'utilisateur et affiche une chaîne de 8 caractères correspondant à son code binaire en complément à 2. (conformément à la convention vue en cours)

Exercices de synthèse

Exercice 9 : **Calendrier**

On considère que la variable suivante, de type `list`, est déjà en mémoire :

```
semaine=["lundi", "mardi", "mercredi", "jeudi", "vendredi", "samedi", "dimanche"]
```

Ecrire un programme qui crée et remplit une liste nommée `octobre`, qui contiendra toutes les dates complètes des 31 jours du mois d'octobre qui commence un samedi :

La liste commence comme ceci :

```
['samedi 1 octobre', 'dimanche 2 octobre', ...(etc)]
```