

TP 2 : Programme et boucle for

Écrire un programme

Comme vous l'avez probablement déjà remarqué, appuyer sur enter après une ligne de commande l'exécute, pour faire un programme, il va falloir s'y prendre autrement.

Pour écrire un programme, il faut ouvrir l'éditeur de programme. Pour ce faire, on sélectionne : Applications, puis Scinotes.

On peut également cliquer sur le bouton  en haut à gauche de l'écran.

La fenêtre qui apparaît est appelée l'éditeur de programme, c'est dans celle ci que l'on peut écrire notre programme.

Une fois le programme rédigé, il ne reste plus qu'à l'exécuter. Pour ce faire, sélectionnez Exécuter, puis Enregistrer et exécuter. On peut également se contenter d'appuyer sur la touche F5.

EXERCICE 1. Si ce n'est déjà fait commencez par créer un dossier ProgrammesScilab dans *Mes documents* où vous enregistrerez vos programmes Scilab et un sous dossier TD2. Prenez l'habitude de faire ça à chaque TD.

Ouvrez ensuite l'éditeur de programme et écrivez le programme suivant :

```
for i=2 :10
    disp(i)
end
```

AVANT de lancer le programme, reprendre la fiche Scilab 2 et prédire le résultat de ce programme.

Puis, lancer le programme et notez ce que vous avez obtenu. Est ce que vous aviez prédit ?

Prédiction :

ça c'est à vous de voir

Résultat de l'algorithme :

2.
3.
4.
5.
6.
7.
8.
9.
10.

EXERCICE 2. Écrire un programme qui demande à l'utilisateur son année de naissance et qui affiche son âge.

```
n=input("Quelle est votre année de naissance ?")
res=2019-n
disp(res)
```

EXERCICE 3. Exprimer le résultat des programmes suivants avec des Σ , prédire alors le résultat avec une formule vue cours, puis vérifiez votre résultat avec Scilab.

`sum(ones(1,7))`

$$\sum_{k=1}^7 1 = 7 - 1 + 1 = 7$$

`sum([1 : 5])`

$$\sum_{k=1}^5 k = \frac{1}{2} \times 5 \times (5 + 1) = 15$$

`sum([1 : 5].^2)`

$$\sum_{k=1}^5 k^2 = \frac{1}{6} \times 5 \times (5+1) \times (2 \times 5 + 1) = 55$$

Écrire ce qu'il faut taper pour calculer $\sum_{k=0}^{11} 2^k$? Écrire ce que répond Scilab et vérifier par une formule vue en cours qu'il s'agit de la bonne réponse.

```
sum(2^[0 :11])
```

4095

$$\sum_{k=0}^{11} 2^k = \frac{1 - 2^{11+1}}{1 - 2} = 2^{12} - 1$$

Vous pouvez vous aider de Scilab pour vous assurer que $2^{12} - 1 = 4095$.

Décrire ce que fait le programme suivant, donner la réponse finale de l'algorithme, mais aussi détailler le fonctionnement de la boucle for à chaque étape.

```
U=1
for i=1 : 11
    U=U+2^i
end
disp(U)
```

Votre réponse :

Ce programme permet de calculer $\sum_{k=0}^{11} 2^k$.

On commence avec le terme $k = 0 : U = 1$

On calcule ensuite pour $i = 1$ le deuxième terme et on l'ajoute à ce qu'on avait précédemment, on trouve $U = 1 + 2^1 = 3$

On calcule ensuite pour $i = 2$ le troisième terme et on l'ajoute à ce qu'on avait précédemment, on trouve $U = 3 + 2^2 = 7$

Et ainsi de suite ...

Jusqu'à 4095

Écrire deux programmes qui demandent à l'utilisateur d'entrer un entier n , qui calculent et enfin affichent $\sum_{k=1}^n \ln(k)$.

L'un en utilisant une boucle for :

```
n=input("Entrer une valeur pour n")
U=0
for i=2 : n
    U=U+log(i)
end
disp(U)
```

L'autre en utilisant la fonction sum :

```
n=input("Entrer une valeur pour n")
sum(log([1 :n]))
```

EXERCICE 4. Décrire ce que fait le programme suivant, donner la réponse finale de l'algorithme, mais aussi détailler le fonctionnement de la boucle for à chaque étape.

```
U=1
for i=1 : 5
    U=U*2
end
disp(U)
```

Votre réponse :

Ce programme permet de calculer $\prod_{k=1}^5 2$.

On commence par un produit vide : $U = 1$

On calcule ensuite pour $i = 1$ le premier terme et on l'ajoute à ce qu'on avait précédemment, on trouve $U = 1 \times 2 = 2$

On calcule ensuite pour $i = 2$ le troisième terme et on l'ajoute à ce qu'on avait précédemment, on trouve $U = 2 \times 2 = 4$

Et ainsi de suite ...

Jusqu'à 32

Écrire deux programmes qui demandent à l'utilisateur d'entrer un entier n , qui calculent et enfin affichent $n!$

L'un en utilisant une boucle for :

```
n=input("Entrer une valeur pour n")
U=1
for i=2 : n
    U=U*i
end
disp(U)
```

L'autre en utilisant la fonction prod :

```
n=input("Entrer une valeur pour n")
prod([1 :n])
```

EXERCICE 5. On considère la suite définie par :

$$u_0 = 0, u_1 = -9 \text{ et } \forall n \in \mathbb{N}, u_{n+2} = 6u_{n+1} - 9u_n$$

Écrire un programme qui demande à l'utilisateur une valeur entière n puis qui affiche en sortie u_n .

```
n=input("Entrer une valeur pour n")
U=0
V=-9
for i = 2 :n
    mem=V
    V=6*V-9*U
    U=mem
end
disp(V)
```

Vous pouvez vérifier que vous retrouvez : $u_6 = -13122$.