

1 Rappels : boucles For

Tester les programmes suivants :

```
--> for i=1:6
> disp("Toto")
> end
```

```
for k=0:5
> disp("Toto")
> end
```

Le choix des lettres i et k dans les deux programmes précédents semble t'il avoir une importance ?

Réponse :

Tester à présent

```
--> for i=1:10
> disp(i+1)
> end
```

```
--> for i=[1,3,5,7]
> disp(i+1)
> end
```

Quelle est la différence entre ces deux programmes ?

Réponse :

Exercice 1

Ecrire un programme qui écrit à la suite les 10 premiers carrés (de 1^2 à 10^2)

Exercice 2

On pose $C = \begin{pmatrix} 2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & \frac{1}{2} \end{pmatrix}$

Ecrire un programme qui affiche à la suite les matrices C, C^2, C^3, C^4, C^5 .

Pouvait-on prévoir ce résultat ? Quelle est la formule générale pour C^m ?

Réponse :

.....

Exercice 3

Sans le taper, prévoir le résultat programme suivant ? Réponse :

```
--> v=1
--> for i=1:6
  > v=v*i;
  > end
--> disp(v)
```

Note : Un ";" évite d'afficher des calculs intermédiaires. Ré-essayer le programme précédent sans le ";" pour suivre les calculs intermédiaires.

Exercice 4

On considère la suite u_n définie par $u_0 = 1$ et $u_{n+1} = 2u_n + 1$ pour tout $n \in \mathbb{N}$.

Que calcule le programme suivant pour cette suite ? Réponse :

```
--> u=1
--> for i=1:6
  > u=2*u+1;
  > end
--> disp(u)
```

Exercice 5

On considère la suite u_n définie par $u_0 = 2$ et $u_{n+1} = -u_n + 3$ pour tout $n \in \mathbb{N}$.

Compléter le programme suivant pour qu'il affiche u_8 :

```
--> u=.....
--> for .....
  > u=.....;
  > end
--> disp(u)
```

Comment modifier le programme suivant pour qu'il affiche, non pas seulement u_8 mais tous les termes de u_1 à u_8 . Que remarquez-vous ?

2 Générer des nombres aléatoirement

Scilab permet de générer des listes de nombres aléatoirement suivant une loi donnée avec la commande `-> grand(m,n,'uin',inf,sup)` qui génère une matrice à m lignes et n colonnes dont les coefficients sont des nombres entiers aléatoires entre inf et sup suivant une loi uniforme.

Exemple : Tester :

```
--> grand(1,10,'uin',1,6)
```

Pour compter combien de fois un résultat est sorti, le 3 par exemple, on utilise la commande `find` :

```
--> tirages=grand(1,10,'uin',1,6) , find(tirages==3)
```

Le logiciel retourne la liste des rangs de la matrice *tirages* où il a trouvé la valeur 3, et la liste vide (`[]`) s'il ne l'a pas trouvée du tout.

Pour savoir finalement combien de fois on a obtenu 3, on peut utiliser la commande `length` qui renvoie le nombre d'éléments d'une liste. Ainsi `length([2,5,2,6,9])` renverra 5 puisque la liste contient 5 élément.

Exercice 6

Comment modifier le programme précédent pour qu'il affiche, non pas les rangs auquel on trouve un 3 mais le nombre de 3 qui sont sortis ?

Ecrire alors un nouveau programme qui affiche la **proportion** de 3 obtenus sur 10 000 lancers. Effectuer plusieurs fois ce programme. Le résultat vous semble t'il conforme ?

Pour afficher les résultats dans un diagramme avec la commande `bar`, tapons le programme suivant dans Scinotes :

```
clf (Nettoie la fenêtre graphique)
tirages=grand(1,10,'uin',1,6) (10 lancers de dé)
resultats=[1,2,3,4,5,6] (Liste des résultats possibles)
for i=1:6
effectifs(i)=length(find(tirages==resultats(i)));
end
bar(resultats,effectifs) (Création du diagramme).
```

Expliquer le rôle de la ligne `effectifs(i)=length(find(tirages==resultats(i)))` ?

On peut garder une trace du diagramme avec *Fichier/Exporter vers* et en le nommant par `figure.jpg` exemple.

Recommencer ensuite en modifiant le nombre de tirages à 100 puis à 1000.

Exercice 7

Rédiger un programme qui affiche un diagramme similaire non plus pour un seul dé, mais pour la somme de deux dés.