

TD 4 langage SCILAB

ESSADDOUKI Mostafa (essaddouki@gmail.com), 10 mai 2018

Exercice 1

Ecrire un programme demandant un entier n à l'utilisateur et affichant la valeur de :

$$a) \sum_{k=0}^n u_k \quad \text{avec} \quad \begin{cases} u_0 = 1 \\ \forall n \in \mathbb{N}, u_{n+1} = \sqrt{u_n + n} \end{cases}$$

$$b) \prod_{k=0}^n u_k \quad \text{avec} \quad \begin{cases} u_0 = 1, u_1 = 2 \\ \forall n \in \mathbb{N}, u_{n+2} = u_{n+1} + nu_n \end{cases}$$

$$c) \sum_{k=0}^n u_k v_k \quad \text{avec} \quad \begin{cases} u_0 = 1, v_0 = 2 \\ \forall n \in \mathbb{N}, u_{n+1} = 3u_n + \sqrt{v_n}, v_{n+1} = \frac{1}{n+1}u_n + v_n \end{cases}$$

```

1 u(1)=1
2 s=1
3 for i=2:n
4     u(i)=sqrt(u(i-1)+(i-1))
5     s=s+u(i)
6 end
7 disp(s)

```

```

1 u(1)=1
2 u(2)=2
3 p=2
4 for i=3:n
5     u(i)=u(i-1)+(i-2)*u(i-2)
6     p=p*u(i)
7 end
8 disp(p)

```

```

1 u(1)=1
2 v(1)=2
3 s=2
4 for i=2:n
5     u(i)=3*u(i-1)+sqrt(v(i-1))
6     v(i)=(1/(i-1))*u(i-1)+v(i-1)
7     s=s+u(i)*v(i)
8 end
9 disp(s)

```

Exercice 2

Ecrire une fonction **function f=fact(n)** qui calcule $n!$

```

1 function f=fact(n)
2     f=1
3     for i=2:n
4         f=f*i
5     end
6 endfunction
7
8 function f=fact_rec(n)
9     if(n==1) then f=1
10    else
11        f=n*fact_rec(n-1)
12    end
13 endfunction

```

Exercice 3

Ecrire une fonction **function p=puissance(x,n)** qui calcule x^n

```

1 function p=puissance(x,n)
2     p=x
3     for i=2:n
4         p=p*x
5     end
6 endfunction

```

Exercice 4

On considère la suite $(U_n)_{n \in \mathbb{N}}$ définie par $u_0 = 1$ et, pour tout n de \mathbb{N}^* , $u_n = \sqrt{n + u_{n-1}}$.

Ecrire une fonction récursive ayant pour nom **suite** qui calcule le terme d'indice **n** de la suite $(U_n)_{n \in \mathbb{N}}$.

```

1 function val=suite(n)
2     if(n==1) then val=1
3     else
4         val=sqrt(n+suite(n-1))
5     end
6 endfunction

```

Exercice 5

On considère la suite $(U_n)_{n \in \mathbb{N}}$ définie par :

$$u_0 = 1, \quad u_1 = 2, \quad \forall n \in \mathbb{N}, \quad u_{n+2} = u_{n+1} \sqrt{n+1} + \min(u_n, n).$$

1. Ecrire un programme qui calcule et affiche la valeur de u_n , l'entier n étant entré par l'utilisateur.

```

1 function val=suite(n)
2     u(1)=1
3     u(2)=2
4     for i=3:n
5         min=u(i-2)
6         if(min>(i-2)) then min=i-2
7         u(i)=u(i-1)*sqrt(i-1)+min
8     end
9     val=u(n)
10 endfunction

```

2. Ecrire un programme qui calcule et affiche la plus petite valeur de n telle que $u_n > A$, le réel A étant entré par l'utilisateur.

```

1 function val=suite(A)
2     u(1)=1
3     u(2)=2
4     i=2
5     while(u(i)<A)
6         i=i+1
7         min=u(i-2)
8         if(min>(i-2)) then min=i-2
9         u(i)=u(i-1)*sqrt(i-1)+min
10    end
11    val=i
12 endfunction

```

Exercice 6

1. Ecrire un programme qui crée un tableau de 20 entiers aléatoires compris entre 0 et 100, qui affiche ce tableau ainsi que le plus petit entier, le plus grand entier de ce tableau et les rangs auxquels ils apparaissent.

```

1 T=grand(1,20,'uin',0,100)
2 min=T(1)
3 pmin=1
4 max=T(1)
5 pmax=1
6 for i=2:20
7     if(T(i)<min) then min=T(i);pmin=i;end
8     if(T(i)>max) then max=T(i);pmax=i;end
9 end
10 disp("min=",min,"_pos=",pmin)
11 disp("max=",max,"_pos=",pmax)

```

2. Ecrire un programme qui crée un tableau de 20 entiers aléatoires compris entre 0 et 10, qui demande un entier x à l'utilisateur et qui affiche le nombre de fois qu'apparaît l'entier x dans le tableau.

```

1 T=grand(1,20,'uin',0,10)
2 x=input("saisir_une_valeur_")
3 nb=0
4 for i=1:20
5     if(T(i)==x) then nb=nb+1;end
6 end
7 disp("nombre_=_",nb)

```

Exercice 7

Ecrire un programme qui calcule le nombre d'entiers pairs et le nombre d'entiers impairs d'une liste déjà remplie avec N entiers.

```

1 function [pair,impair]=compteur(T)
2     N=length(T)
3     pair=0
4     impair=0
5     for i=1:N
6         if(modulo(T(i),2)==0) then pair=pair+1;
7         else     impair=impair+1
8     end
9 endfunction

```

Exercice 8

Écrivez un programme qui analyse un par un tous les éléments d'une liste de nombres t1 pour générer deux nouvelles listes. L'une contiendra seulement les nombres pairs de la liste initiale, et l'autre les nombres impairs. Par exemple, si la liste initiale est t1= [32, 5, 12, 8, 3, 75, 2, 15], le programme devra construire une liste paire qui contiendra [32, 12, 8, 2], et une liste impaire qui contiendra [5, 3, 75,15].

```

1 function [pair,impair]=compteur(T)
2     N=length(T)
3     npair=1
4     nimpair=1
5     for i=1:N
6         if(modulo(T(i),2)==0) then pair(npair)=T(i);npair=npair+1
7         else     impair(nimpair)=T(i);nimpair=nimpair+1;end
8     end
9 endfunction

```

Exercice 9

Soit une liste contenant n entiers rangés dans l'ordre croissant.

Ecrire une fonction **function inserer(L,e)** qui insère un nouvel élément e dans la liste en respectant l'ordre croissant.

```
1 function inserer(L,e)
2     N=length(T)
3     while(T(N)>e)
4         T(N+1)=T(N)
5         N=N-1
6     end
7     T(N+1)=e
8 endfunction
```

Exercice 10

Soit une liste T1 d'entiers de taille n . Un plateau dans T1 est une suite d'éléments consécutifs et égaux de T1. Le problème consiste à trouver le plus long plateau dans une liste T1.

Par exemple, pour la liste contenant les valeurs [2 2 6 6 6 6 3 3 6 3 3 3 2 3 3], le plus long plateau commence à l'indice 3 et il est de longueur 4.

Ecrire un programme qui permet de résoudre le problème énoncé.

```
1 function [indice, longueur]=plateau(T)
2     N=length(T)
3     indice=1
4     longueur=1
5     deb=1
6     taille=1
7     for i=2:N
8         if(T(i)==T(i-1))then taille=taille+1
9         else
10            if(taille>longueur)then
11                indice=deb
12                longueur=taille
13            end
14            deb=i
15            taille=1
16        end
17    end
18 endfunction
```