

# TD 3 corrigé : les suites

ESSADDOUKI Mostafa (essaddouki@gmail.com), 9 mai 2018

## Exercice 1

Écrire un programme qui permet de calculer  $S_1$ ,  $S_2$ ,  $S_3$  et  $S_4$  tel que :

- $S_1 = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{N}$
- $S_2 = 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{6} + \dots + \frac{1}{N}$
- $S_3 = 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{6} + \dots - \frac{1}{N}$
- $S_4 = 1 + x + x^2 + \dots + x^N$

```

1 N=input("saisir_N_")
2 s1=1
3 s2=1
4 s3=1
5 s4=1
6 for i=2:N
7     s1=s1+(1/i)
8 end
9 disp("S1_=",s1)
10
11 for i=2:2:N
12     s2=s2+(1/i)
13 end
14 disp("S2_=",s2)
15
16 sign=-1
17 for i=2:2:N
18     s3=s3+sign*(1/i)
19     sign=sign*(-1)
20 end
21 disp("S3_=",s3)
22
23 x=input("saisir_x")
24 for i=1:N
25     s4=s4+(x^i)
26 end
27 disp("S4_=",s4)

```

## Exercice 2

Ecrire un programme qui détermine le 20<sup>ième</sup> terme d'une suite définie par :

$$\begin{cases} S_0 = 2 \\ S_1 = 3 \\ S_n = S_{n-1} + (-1)^n * S_{n-2} \end{cases}$$

```

1 s1=2;
2 s2=3;
3 for i=2:19
4     s=s2+((-1)^i)*s1
5     s1=s2
6     s2=s
7 end
8 disp("le_20_eme_terme_est:", s2)

```

### Exercice 3

Soit la suite  $U$  définie par :

$$\begin{cases} U_0 = \text{est un entier positif pris au hasard (avec } 3 < U_0 < 40) \\ U_n = \frac{U_{n-1}}{2} \text{ si } U_{n-1} \text{ est pair} \\ U_n = 3 * U_{n-1} + 1 \text{ sinon (} n > 0) \end{cases}$$

Cette suite aboutit au cycle redondant formé par les trois termes **4,2,1** à partir d'un certain rang. -

#### Exemple :

Pour  $U_0 = 3$

$U_1 = 10; U_2 = 5; U_3 = 16; U_4 = 8; U_5 = 4; U_6 = 2; U_7 = 1; U_8 = 4; U_9 = 2; U_{10} = 1,$

Donc la suite  $U$  entre dans le cycle redondant 4,2,1 à partir du 6ème terme(rang=6)

Ecrire un programme permettant de déterminer le rang à partir duquel la suite  $U$  aboutit au cycle redondant 4, 2 et 1

- La fonction **grand("uin",3,40)** permet de générer un nombre entier entre 3 et 40

```

1 U=grand("uin", 3, 40)
2 it=1
3 while(U<>4)
4     it=it+1
5     if(modulo(U, 2)==0) then
6         U=U/2
7     else
8         U=3*U+1
9     end
10 end
11 disp("la_suite_U_entre_dans_le_cycle_redondant_a_partir_du_rang=", it)

```

### Exercice 4

On veut obtenir une bonne approximation du nombre  $e$  (qui est la valeur de la fonction exponentielle de 1). Pour cela, on construit une suite de nombres réels de cette façon :  $U_0 = 1$  et pour chaque entier  $n > 0$ ,  $U_n = U_{n-1} + (1/n!)$ . Avec  $n! = 1 \times 2 \times \dots \times n$  Cette suite va tendre vers le nombre  $e$  quand  $n$  tend vers l'infini ( $+\infty$ ).

Ainsi,  $U_1 = U_0 + (1/1!) = 1 + 1 = 2$  et  $U_2 = U_1 + (1/2!) = 2 + (1/2) = 2,5\dots$  Il faut savoir que le terme  $(1/n!)$  va tendre rapidement vers zéro et donc pour une certaine valeur de  $n$ , nous aurons  $U_n = U_{n-1}$  du

fait des imprécisions dans la représentation des nombres réels dans l'ordinateur.

Ecrire un programme qui permet de calculer et d'afficher les valeurs de  $e(= U_n)$  du premier  $n$  qui vérifie  $Un = U_{n-1}$ .

```

1 U1=1
2 U2=U1+1
3 n=2
4 f=1
5 while (U1 <> U2)
6     f=f*n
7     n=n+1
8     U=U+(1/f)
9     U1=U2
10    U2=U
11 end
12 disp(U2)

```

## Exercice 4

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

$$\begin{aligned}
 a) & \sum_{k=0}^n u_k \quad \text{avec} \begin{cases} u_0 = 1 \\ \forall n \in \mathbb{N}, u_{n+1} = \sqrt{u_n + n} \end{cases} \\
 b) & \prod_{k=0}^n u_k \quad \text{avec} \begin{cases} u_0 = 1, u_1 = 2 \\ \forall n \in \mathbb{N}, u_{n+2} = u_{n+1} + nu_n \end{cases} \\
 c) & \sum_{k=0}^n u_k v_k \quad \text{avec} \begin{cases} u_0 = 1, v_0 = 2 \\ \forall n \in \mathbb{N}, u_{n+1} = 3u_n + \sqrt{v_n}, v_{n+1} = \frac{1}{n+1}u_n + v_n \end{cases}
 \end{aligned}$$

```

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

```

```

1 u1=1
2 u2=2
3 p=2
4 for i=2:n
5     u=u2+(i-2)*u1
6     u1=u2
7     u2=u
8     p=p*u

```

```
9 end
10 disp(p)
```

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