

La suite de Fibonacci

Partie A

1) On considère la suite de Fibonacci définie par $F_0 = 0$, $F_1 = 1$

et pour tout $n \geq 2$, $F_n = F_{n-1} + F_{n-2}$

Ecrire en Scilab une fonction qui, pour un entier n donné, calcule la valeur du terme F_n de la suite de Fibonacci : fonction [f]=Fibonacci(n);

Partie B

On désire pouvoir calculer exactement, pour $2 \leq n \leq 100$, la valeur d'un terme F_n de la suite de Fibonacci. La fonction précédente renvoie un résultat erroné à partir de $n = 79$.

Afin de calculer F_n , pour $79 \leq n \leq 100$, sans erreur de troncature ou d'arrondi, on définit l'algorithme suivant :

Cet entier est représenté par un tableau de taille 25 à raison d'un chiffre par élément. Si on note t une variable de type entier, alors $t(25)$ est le chiffre des unités de cet entier, $t(24)$ celui des dizaines, $t(23)$ celui des centaines, etc ... Au delà du dernier chiffre de l'entier, les éléments du tableau sont nuls.

Ainsi $F_{47} = 2971215073$ est représenté par le tableau

0	0	0	0	...	0	2	9	7	1	2	1	5	0	7	3
1	2	3	4	...	15	16	17	18	19	20	21	22	23	24	25

Ce type permet donc de représenter tout entier naturel de l'intervalle $[0 \dots (10^{26} - 1)]$.

2) Ecrire une fonction pour calculer la somme de deux nombres de type entier

fonction [f]=Somme(f1,f2)

Où la somme des entiers représenté par f1 et f2 est donné par la variable f3.

Exemple :

f1	<table border="1" style="display: inline-table; text-align: center;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>...</td><td>0</td><td>8</td><td>1</td><td>7</td></tr></table>	0	0	0	0	0	...	0	8	1	7
0	0	0	0	0	...	0	8	1	7		
	+										
f2	<table border="1" style="display: inline-table; text-align: center;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>...</td><td>1</td><td>4</td><td>6</td><td>4</td></tr></table>	0	0	0	0	0	...	1	4	6	4
0	0	0	0	0	...	1	4	6	4		
	=										
f	<table border="1" style="display: inline-table; text-align: center;"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>...</td><td>2</td><td>2</td><td>8</td><td>1</td></tr></table>	0	0	0	0	0	...	2	2	8	1
0	0	0	0	0	...	2	2	8	1		

3) Ecrire une fonction [f]=Fibonacci2(n); qui construit le tableau t représentant le nombre de Fibonacci F_n .

4) Ecrire une fonction pour afficher à l'écran l'entier naturel représenté par un tableau t de type entier.

Par exemple, pour le tableau f de la question 2), cette procédure devra afficher 2281.

Partie C

5) Ecrire un programme permettant de saisir au clavier la valeur d'un entier n , si cet entier est inférieur à 79, d'utiliser la fonction Fibonacci de la partie A, si l'entier est entre 79 et 100, d'utiliser la fonction Fibonacci2 de la partie B et si l'entier est supérieur à 100, demander un autre entier.

Il suffira d'afficher la valeur du terme F_n ainsi obtenu.

La suite de Fibonacci : Correction

```
function [f]=fibonacci(n)
```

```
u=0;
v=1;
for i=2:n do
f=u+v;
u=v;
v=f;
end
```

```
function [c]=somme(a,b)
```

```
c=zeros(1,25);
for i=25:-1:1 do
c(i)=a(i)+b(i)+c(i);
if c(i)>9 then
c(i)=c(i)-10;
c(i-1)=1;
end
end
```

```
function afficher(f)
```

```
i=1;
while (f(i)==0)&(i<25) do i=i+1; end
c="";
for j=i:25 do c=c+string(f(j)); end
disp(c)
```

```
function [f]=fibonacci2(n)
```

```
u=zeros(1,25);
v=zeros(1,25);v(25)=1;
for i=2:n do
f=somme(u,v);
u=v;
v=f;
end
afficher(f);
```

```
function programme;
```

```
n=input('entrez un entier')
while n>100 do n=input('entrez un entier'); end
if n<79 then disp(fibonacci(n)); else fibonacci2(n); end
```