

# 1 Fonctions

On rappelle la syntaxe de la définition d'une fonction avec Scilab :

```
function variable_de_sortie = nom_de_la_fonction (variable_d'entrée)

    (Série d'instructions permettant le calcul de la valeur
    de la variable de sortie)

endfunction
```

### Exemple

Définissons la fonction  $f : x \mapsto x^2 + x + 1$  que nous appellerons trinome :

```
function y = trinome (x)
    y=x^2+x+1
endfunction
```

Dans la console, on peut ensuite appeler la fonction :

```
--> trinome(2)
ans=7
```

### Exercice 1

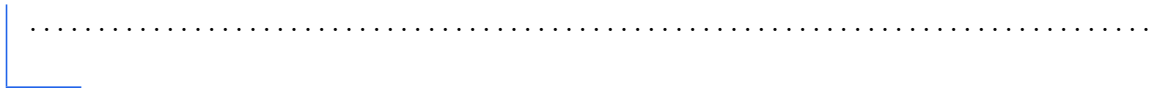
- (a) Définir la fonction Cblanc définie par :  $C(x) = (x^2 + 1)e^{-x}$ .
- (b) Utiliser cette fonction pour déterminer une valeur approchée de  $C(1)$  et de  $C(0,5)$ .

.....  
.....

### Exercice 2

- (a) Soit  $(v_n)$  la suite définie par  $v_n = \frac{3n^2}{2n^2 + 1}$ . Écrire une fonction **suite\_v** qui à tout nombre  $n$  associe la valeur de  $v_n$ .
- (b) Calculer *suite\_v*(20)
- (c) Programmer une boucle qui permet d'afficher  $v_0, v_1, \dots, v_{50}$  Que semble valoir  $\lim_{n \rightarrow \infty} v_n$ ?  
Pouvait-t'on prévoir cette valeur?

.....  
.....



## 2 Scilab : Instruction conditionnelle “if then else”

Rappel de la syntaxe :

```
if condition then instruction
    else instruction
end
```

**Remarque** : la ligne commençant par *else* n'est pas obligatoire.

### Exemple 1

On veut tester si un nombre  $x$  fourni par l'utilisateur est inférieur ou égal à 20 :

```
x=input("Nombre : ")
if x<=20 then rep="oui"
    else rep="non"
end
disp(rep)
```

### Exemple 2

On veut tester si un nombre  $n$  fourni par l'utilisateur est un nombre pair :

```
n=input("Nombre : ")
if n\%2==0 then rep="Le nombre est pair"
    else rep="Le nombre est impair"
end
disp(rep)
```

### Exercice 3

- Écrire un programme qui teste si un nombre entier fourni par l'utilisateur est un multiple de 3.
- Écrire un programme qui teste si un nombre fourni par l'utilisateur est compris entre 0 et 20 au sens large.

#### Exercice 4

La célèbre suite de Syracuse est définie par la donnée d'un premier terme  $u_0$  entier naturel non nul, et par la relation de récurrence suivante :

$$\begin{cases} u_{n+1} = \frac{u_n}{2} \text{ si } n \text{ est pair ;} \\ u_{n+1} = 3u_n + 1 \text{ si } n \text{ est impair.} \end{cases}$$

- (a) Calculer à la main les vingt premiers termes de la suite dans le cas où  $u_0 = 7$ .

Une conjecture encore non démontrée à ce jour énonce que, quelle que soit la valeur de  $u_0$ , la suite finit toujours par prendre la valeur 1.

- (b) Écrire un programme scilab qui détermine, à partir de la valeur de  $u_0$  fournie par l'utilisateur, le premier rang  $n$  pour lequel  $u_n = 1$ .  
Appliquer ce programme à  $u_0 = 101$