É.Bouchet ECS1 # TP3 - Boucles if et while dans Scilab

Attention : pour l'ensemble de ce TP, si vous avez l'impression que votre console ne répond plus, entrez « end » autant de fois que nécessaire pour compenser les fins de boucle que vous auriez pu oublier (et vérifiez votre code avant de recompiler).

1 Les boucles if

1.1 Syntaxe

Les **boucles if** servent aux disjonctions de cas. La syntaxe Scilab est la suivante :

```
if expression1 then instructions
elseif expression2 then nouvelles instructions
...
else dernières instructions
end
```

Exercice 1. Déterminer ce que fait la fonction définie comme suit :

```
function y = f(a,b);
 if a > b then y = a ;
 else y = b ;
 end
endfunction
```

1.2 Connecteurs logiques

Dans l'expression de la boucle if, on peut utiliser tous les comparateurs et éléments logiques suivants :

- `==` pour tester l'égalité de deux valeurs (attention, il est nécessaire de mettre les *deux* signes, comme le `=` est déjà utilisé pour l'affectation des variables)
- `>` ou `<` pour tester si l'une est strictement inférieure à l'autre.
- `>=` ou `<=` pour tester des inégalités larges.
- `<>` pour tester la différence de deux valeurs.
- `&` pour le connecteur logique « et ».
- `|` pour le connecteur logique « ou » (maj+alt+l sous mac).

Exercice 2. Déterminer ce que fait la fonction définie comme suit :

```
function y = test(a,b,c)
 if ( (a==b) | (b==c) | (a==c) ) then y=1;
 else y=0;
 end
endfunction
```

Exercice 3. Écrire une fonction qui prend en entrée un réel, qui renvoie "oui" si ce réel est un entier, et "non" sinon.

1.3 Autres utilisations des booléens

En informatique, on appelle booléen un type de variable à deux états (vrai et faux), destiné à représenter les valeurs de vérité. Les conditions utilisées dans les boucles if sont en particulier des booléens. Scilab utilise `T` pour vrai et `F` pour faux.

Exercice 4. Donner un exemple de commande Scilab simple qui renvoie un booléen.

Les booléens s'intègrent assez naturellement à certaines fonctions usuelles.

Exercice 5. Expliquer ce que renvoient les programmes suivants.

1. ```
   A = 1:10 ;
   A > 3
   ```
2. ```
   sum(A > 3)
   ```

2 Les boucles while

Les boucles for reproduisent une opération un nombre déterminé de fois. Mais parfois, on préfère reproduire l'opération tant qu'une condition n'est pas réalisée, sans savoir combien d'itérations seront nécessaires. Pour cela, on utilise des **boucles while**. La syntaxe Scilab est la suivante :

```
while expression
instructions
end
```

Remarque. Il faut faire particulièrement attention en utilisant des boucles while : si la condition reste toujours vérifiée, la boucle continue de tourner à l'infini et finit par bloquer le logiciel.

Exercice 6. Écrire un programme contenant une boucle while qui permet de trouver le plus petit entier $n \geqslant 1$ tel que $\dfrac{1}{n} \leqslant 0,01$.

Exercice 7. Écrire une fonction `test` qui prend en argument un réel M et qui renvoie la plus petite valeur de $n!$ telle que $n! \geqslant M$.