

TP2 : Programmation – if then else

Un rappel approximatif sur la programmation

Un programme informatique *est* une série d'instruction qui seront exécutées par l'ordinateur. Dans Scilab, on peut entrer une suite d'instruction directement dans la console, ou bien dans un document spécifique, enregistré au format `.sce` et appelé dans la console par **exec nom_du_fichier.sce**. Un programme peut contenir des *fonctions* qui, à l'instar des fonctions mathématiques, prennent une ou plusieurs variables en entrée et, après une série d'instructions, renvoie une ou plusieurs variables en sortie.

Exercices d'écriture de fonctions

Q.1 Écrire, dans SciNote, une fonction appelée **convTempFtoC** qui prend en argument une température en degré Fahrenheit et renvoie la température en degré Celsius. On rappelle que si x est la température en degré Celsius, alors $1.8x + 32$ est la température en degré Fahrenheit.

On veillera à ne pas tomber dans le "piège"...

Q.2 Écrire une fonction appelée **convEURtoUSD** qui prend en argument un montant en euro et renvoie sa valeur en dollars. On considérera que $1\text{€} = 1.11739\text{\$}$ (cours moyen, 2016-09-16 14:56 UTC).

Dialogue avec l'utilisateur

On a utilisé à la fin du TP1, la fonction **input** qui permet de demander une valeur à l'utilisateur. Consulter à nouveau l'aide pour comprendre comment utiliser la fonction **input**. On utilise la fonction **disp** pour faire afficher une variable dans un programme.

Q.3 Dans une nouvelle fenêtre SciNote, créer un programme qui demande à l'utilisateur son âge n et qui affiche "Vous avez n ans". Consulter l'aide de la commande **disp**.

Pour concaténer (mettre à la suite) des chaînes de caractères, on peut utiliser l'opérateur `+`. Ainsi, au lieu d'avoir une sortie en 3 lignes pour le programme précédent, on aurait pu écrire

```
disp("Vous avez "+string(n)+" ans")
```

où la commande **string()** transforme la valeur d'une variable en chaîne de caractère.

Structures conditionnelles

Pour élaborer des programmes plus sophistiqués, on a rapidement besoin de structures conditionnelles. Considérons un exemple. Nous savons demander à l'utilisateur son âge et souhaitons maintenant lui indiquer s'il pourra voter aux élections présidentielles de 2017. Nous avons donc besoin de distinguer les réponses en deux catégories : moins de 18 ans (pour simplifier) et plus de 18 ans. En pseudo-langage, notre algorithme pourrait s'écrire

```

n = Âge indiqué par l'utilisateur
Si n<=18 Alors
    Afficher "Vous pourrez voter en 2017"
Sinon
    Afficher "Vous ne pourrez pas voter en 2017"

```

En Scilab, comme dans tous les langages de programmations, on utilise les mots clés **if** **then** et **else**. Pour effectuer une comparaison, on utilise les symboles du tableau suivant. Pour exprimer plusieurs conditions, on utilisera également les opérateurs suivants.

Test	Scilab
=	==
≤	<=
≥	>=
<	<
>	>
≠	<>

Opérateur	Scilab
ET	&
OU	

Q.4 Écrire un programme correspondant à l'algorithme décrit ci-dessus.

Q.5 Prévoir le résultat des suites d'instructions suivantes et vérifier avec Scilab¹ :

```

x=3
y=2
if x<>y then
    disp("x et y sont distincts")
else
    disp("x et y sont égaux")

x=10
y=4
if ( x==3 | y<=5) & (x<9 | y<>4) then
    disp("C"était vrai")
else
    disp("C"était faux")

x=2
y=5
if x<3 & y<=4 then
    disp("C"était vrai")
else
    disp("C"était faux")

```

Pour simplifier l'affichage (ne pas afficher le code du programme), on utilisera l'instruction `exec('nom_du_fichier.sce',-1)` pour exécuter les programmes.

Q.6 Écrire un programme qui demande à l'utilisateur de donner un réel x et qui affiche $\ln(x)$ si $x > 0$ et "Le logarithme n'est pas défini" sinon.

Q.7 Écrire un programme qui demande deux réels a et b à l'utilisateur et qui affiche les éventuelles solutions de l'équation $ax + b = 0$.

1. Pour afficher une apostrophe, on doit signifier à Scilab qu'il ne s'agit pas d'une fin de chaîne de caractères : on écrit " pour '.

Q.8 Écrire un programme qui demande trois réels a , b et c à l'utilisateur et qui affiche les éventuelles solutions de l'équation $ax^2 + bx + c = 0$.

Lorsque l'on souhaite effectuer des tests plus complexes, on peut utiliser le mot-clé **elseif** pour lequel on consultera l'aide.

Q.9 Écrire un programme, utilisant la fonction **rand**², affichant "pierre", "feuille" ou "ciseau" avec probabilité 1/3.

Q.10 Écrire un programme permettant la conversion de devises à partir du tableau suivant :

	USD	EUR	GBP
USD	1	0.89583	0.76450
EUR	1.11616	1	0.85331
GBP	1.30804	1.17191	1

FIGURE 1 – Taux de change. 2016-09-16 15 :01 UTC (Source : xe.com)

Votre programme pourra demander à l'utilisateur quelle est sa devise courante puis quelle est la devise vers laquelle il souhaite faire la conversion.

2. qui renvoie un réel pris au hasard entre 0 et 1.