TP 6 : Structures conditionnelles

Pré-requis : avant d'entamer ce TP, il faut avoir lu / compris / effectué les manipulations présentes dans le cours « Chapitre 6 : Structures conditionnelles » ainsi que les précédents.

- Dans le dossier `Info_1a`, créez le dossier `TP_11`.

I. Écrire des conditionnelles

I.1. Écrire des tests à l'aide d'opérateurs de comparaison

- Compléter le tableau suivant en y notant, dans un premier temps, vos prédictions.
  Ceci étant fait, évaluer et comparer le résultat avec vos prédictions.

Commandes	`8 == 3`	`8 <> 3`	`8 < 3`	`8 > 3`	`8 <= 8`	`8 >= 8`	$\sim$ `(8 == 3)`
Prédiction							
Évaluation							

- Évaluer `((3 >= 8) | ~(6 >= 8)) & (3 <> 4) & (%T | %F)` et expliquer le résultat obtenu.

I.2. Écriture de structures conditionnelles

On considère les fonctions suivantes.

```
// Une structure conditionnelle ...
function y = Test1(x)
    if x > 3 then
        y = 2
    elseif x < 14 then
        y = 8
    end
endfunction
```

```
// ...et une autre
function y = Test2(x)
    if x > 3 then
        y = 2
    elseif x > 14 then
        y = 8
    end
endfunction
```

- Recopier ces fonctions dans **SciNotes** puis enregistrer et exécuter.

- Décrire très précisément les différents éléments de ce code (nom de la fonction, nom des arguments en entrée, nom des arguments en sortie).

- Compléter le tableau suivant en y notant, dans un premier temps, vos prédictions.
  Ceci étant fait, évaluer et comparer le résultat avec vos prédictions.

Appels	`Test1(8)`	`Test1(16)`	`Test1(0)`	`Test1(3)`
Prédiction				
Évaluation				

- Compléter le tableau suivant en y notant, dans un premier temps, vos prédictions.
  Ceci étant fait, évaluer et comparer le résultat avec vos prédictions.

Appels	`Test2(8)`	`Test2(16)`	`Test2(0)`	`Test2(3)`
Prédiction				
Évaluation				

- Comment peut-on corriger le code de la fonction `Test2` afin de ne plus obtenir de message d'erreur ?

II. Exercices d'application

II.1. Exercice 1

- Écrire la fonction :
  - × dont le nom est `absolu`,
  - × prenant un argument d'entrée nommé `x`,
  - × utilisant un argument de sortie nommé `y`,
  - × qui calcule la valeur de `|x|` et la stocke dans `y`.

  Il est évidemment interdit, pour cette question, d'utiliser la fonction prédéfinie `abs`.

- Afin de tester la fonction précédente de manière ludique, écrire le programme qui :
  - × demande à l'utilisateur d'entrer une valeur au clavier et la stocke dans une variable nommé `x`,
  - × à l'aide d'une structure conditionnelle :
    - ***a)*** teste l'égalité de l'appel `abs(x)` et de l'appel `absolu(x)`,
    - ***b)*** affiche `Je suis le king de Scilab` si le test précédent est vérifié,
    - ***c)*** affiche `Va corriger ta fonction absolu !` si le test précédent n'est pas vérifié.

- Exécuter ce programme en choisissant successivement les valeurs suivantes pour `x` :
  -3, $-\mathrm{e}\sqrt{2}$, 0, 5, $\ln(\lfloor 3.7 \rfloor)$

II.2. Exercice 2

Notons r est l'unique racine réelle du polynôme : $P(x) = 5\ x^3 + 1$.
On considère la fonction f suivante.

$$f \ : \left| \begin{array}{ccc} \mathbb{R} & \to & \mathbb{R} \\ x & \mapsto & \left\{ \begin{array}{ll} 0 & \text{si } x \leqslant r \\ 5\ x^3 + 1 & \text{si } r < x \leqslant 0 \\ \dfrac{\mathrm{e}^x}{x+1} & \text{si } x > 0 \end{array} \right. \end{array} \right.$$

- Écrire la fonction :
  - × dont le nom est `f`,
  - × prenant un argument d'entrée nommé `x`,
  - × utilisant un argument de sortie nommé `y`,
  - × qui calcule la valeur de $f($`x`$)$ et la stocke dans `y`.

- Dans la console **Scilab**, effectuer les appels suivants : $f(-3)$, $f(-\mathrm{e}\sqrt{2})$, $f(0)$, $f(5)$, $f(\ln(\lfloor 3.7 \rfloor))$. Notez ci-dessous les valeurs obtenues.

III. Codage d'un jeu aléatoire

On considère le jeu suivant qui permet de gagner ou perdre de l'argent en fonction du résultat d'un lancer de trois dés. Le gain ou la perte sont donnés par les conditions suivantes :

- si le résultat du lancer est un triple 6, le gain est de 10 €.
- si les trois dés sont identiques (différents de 6), le gain est de 5 €.
- si seulement deux dés sont identiques, le gain est de 1 €.
- sinon, la perte est de 5 €.

▶ Exécutez plusieurs fois la commande `grand(1,3,"uin",1,6)`. À quoi sert-elle ?

▶ Comparez les résultats obtenus avec votre voisin. Commentez.

- Écrire la fonction :
  - × dont le nom est `gainJeu`,
  - × ne prenant aucun argument en entrée,
  - × qui simule le lancer de 3 dés,
  - × qui calcule le gain (éventuellement négatif) relatif à ce lancer et stocke le résultat dans la variable `gain`, argument de sortie de la fonction.

- Écrire un programme :
  - × qui réalise un appel à la fonction `gainJeu`,
  - × qui, en cas de perte, affiche : `Encore perdu ! Et encore 5 euros de moins !`
  - × qui, en cas de gain, affiche : `Enfin un gain ! Voici tes x euros`.
    La variable `x` doit être remplacée par sa valeur.

- Dénombrez les ensembles correspondant à chacun des critères de gain.
  Pensez-vous qu'il est intéressant de jouer à ce jeu ?

IV. Codage du prix d'une commande

Le CDI d'un lycée souhaite commander plusieurs exemplaires d'un livre de mathématiques. Le commerçant vendant ce livre propose un prix dégressif :

- 32 € par livre pour toute commande de 1 à 5 livres,
- 30 € par livre pour toute commande de 6 à 30 livres,
- 25 € par livre pour toute commande de 31 à 50 livres,
- 22 € par livre pour toute commande de 51 livres ou plus.

▶ Écrire la fonction :

× dont le nom est `prixCommande`,

× prenant un argument d'entrée nommé `nbL` représentant le nombre de livres de la commande,

× qui calcule le prix de la commande et stocke le résultat dans la variable `prix`, argument de sortie de la fonction.