PRÉSENTATION DE MATLAB
TEST IF – ELSE - END

Le logiciel Matlab est à la fois un langage de programmation et un logiciel de calcul numérique.

La puissance de Matlab vient des très nombreuses fonctions préprogrammées (tracés de courbes, inversion de matrice, résolution d'équations différentielles, …) permettant de résoudre divers problèmes numériques : algèbre linéaire, intégration numérique, optimisation, statistique,…

Mais cette année nous allons nous contenter de découvrir la syntaxe permettant le calcul numérique et d'apprendre à programmer avec quelques instructions simples.

CALCULATRICE ET AIDE EN LIGNE :

On peut utiliser MATLAB comme une simple calculatrice :

1. Lancer MATLAB. Le prompt >> apparaît, indiquant que Matlab attend une instruction.

2. Taper et commenter : `>> 1+1` `>> 1+9*8` `>> 3^2` `>> 1/3*3` `>> 1/3` puis `ans*3` `>> sin(0)` `>> sin(pi/2)`.

3. Il est possible de taper plusieurs instructions sur la même ligne en les séparant par des virgules. Taper par exemple : `>> 1+1, 2+2, 4+4, 8+8`

4. Taper : `>> 1+2` `>> 1+2+3` `>> 1+2+3+4` `>> 1+2+3+4+5`

5. **Astuce :** utiliser la flèche vers le haut ↑ pour refaire la séquence précédente plus rapidement.

6. Ne pas hésiter à utiliser l'aide de MATLAB (aide en ligne). Pour y faire appel taper simplement `help` suivi du nom de la fonction. Découvrez la fonction `log` (`help log`) puis calculer ln(2) puis log(2).

7. Grâce à `help elfun` (elementary function) noter la syntaxe pour prendre la valeur absolue d'un nombre puis pour sa partie entière (faire la différence entre `floor`, `ceil`, `round` et `fix`).

8. MATLAB possède une liste de nombres pseudo-aléatoires compris entre 0 et 1 exclus. La commande `rand` permet de piocher un nombre dans cette liste. Taper plusieurs fois : `>> rand`.

9. Produire un nombre réel compris aléatoirement entre 0 et 5. Puis entre -4 et 10.

10. Enfin produire un **entier** compris entre 1 et 6 (modélisation du tirage d'un dé).

AFFECTATION DE VARIABLES :

Il est possible de créer des variables en leur affectant une valeur numérique. On peut ensuite manipuler ces variables pour des calculs littéraux.

11. Taper : `>> a = 2` `>> b = 3^2` `>> c = a+2` `>> d = 2*a + b^2`

Le signe = en informatique n'a pas la même signification qu'en maths. Il s'agit d'un outil d'affectation : la variable de gauche reçoit la valeur du terme de droite.

12. Taper et méditer : `>> a = a+1` (incrémentation de la variable `a`) `>> b = b*2+a`

Attention : une même variable peut apparaître à gauche et à droite du signe égale (a = a+1) : la nouvelle valeur de cette variable (terme de gauche) est calculée à partir de l'ancienne valeur (terme de droite).

13. Taper : >> clear (effacement de la mémoire) puis >> a = a + 1. Retenir le message.

Attention : une incrémentation doit toujours être précédée d'une initialisation.

Attention : le nom des variables ne doit pas contenir d'espace, ni d'accents, ni de caractères spéciaux : % @ ^ ( ) & , …

AFFICHAGE DES VARIABLES :

Pour afficher la valeur d'une variable, il suffit de taper son nom après le prompt.

14. Taper : >> pi >> a (interpréter !) >> a = 3 ; a = a + 2 ; >> a

On remarque qu'une instruction est suivie d'un point-virgule, MATLAB n'affiche pas le résultat à l'écran. Cette remarque est intéressante quand les calculs sont nombreux et que seul le résultat final est intéressant.

15. Noter la différence entre : >> b = 5, b = b+3, b=b+3, b=b+3

et >> b = 5 ; b = b+3 ; b=b+3 ; b=b+3

16. Il est possible d'afficher une chaîne de caractère en l'écrivant entre apostrophes. Taper : >> 'Bonjour' >> 'vive l'informatique' >> 'vive l''informatique'

SAISIE PAR L'UTILISATEUR :

La valeur de certaines variables peut ne pas être fixée lors de la programmation mais être demandée à l'utilisateur du programme. Il suffit d'utiliser la fonction input.

17. Taper : >> x = input('Tapez votre âge : '). MATLAB attend la saisie d'une valeur numérique qu'il stockera dans la variable x.

TEST :

MATLAB réalise des tests grâce aux opérateurs de comparaison qui sont < , > , <= , >=, ~= (qui signifie « différent ») et == **(et non = qui sert à l'affectation)**.

Les opérateurs logiques sont | (ou), & (et), ~ (non).

Lors d'un test Matlab renvoie 0 quand le test est faux ou 1 si le test est vrai.

18. Taper les tests suivants et observer la réponse de MATLAB :

>> a = 2, b = 3 >> a < 3 >> a < 1 >> a == b

>> a ~= 4 >> b <= 3 >> a < 3 & b > 2 >> a < 3 | b < 3

Attention : ne pas confondre a = b (affectation) avec a == b (test). Ainsi 2 == a est possible mais pas 2 = a.

FONCTION IF – ELSE - END :

Les tests sont utilisés avec la fonction IF qui rend l'exécution d'une instruction conditionnelle.

Les instructions comprises entre le test et le end ne sont exécutées que si le test est vérifié.

Structure :
```
if test
   instruction_1
   instruction_2
end
```

19. Exemple : `>> x = input('taper votre âge : ') ; if x >= 18, 'vous êtes majeur',end`

Il est possible de compléter la fonction IF – END par l'ajout de ELSE : les instructions après le `else` ne sont exécutées que si le test est faux (test = 0).

Structure :
```
if test
   instruction_vrai
else
  instruction_faux
end
```

Attention : Il n'y a pas de test après ELSE puisque le test a déjà été fait lors du IF.

20. Compléter la ligne 19 pour que MATLAB affiche 'vous êtes mineur' lorsque c'est le cas.

21. Il est possible d'emboîter plusieurs `if else end`. Voici des instructions qui renvoient le signe de la variable x (-1 pour $x < 0$, 0 pour $x = 0$ et +1 pour $x > 0$).

```
x = input('saisir un nombre ') ;
if x < 0
   y = -1,
else
   if x == 0
      y = 0,
   else
      y = 1,
   end,
end
```

EXERCICES :

22. Taper une ligne d'instruction demandant à l'utilisateur de saisir sa dernière note et affichant « très bien » si la note supérieure à 13, « bien » entre 8 et 13 et « courage » si la note est inférieure à 8.

23. Taper une ligne d'instruction qui résout $a.x+b = 0$ dans $\mathbb{R}$ où a et b sont deux réels saisis par l'utilisateur. La tester avec (2,3), (2,0), (0,2) et (0,0).

24. Faire une fonction qui demande à l'opérateur de saisir un nombre x entre 1 et 1000, qui vérifie que la saisie est correcte ($x \in [0, 1000]$), puis, si la saisie est correcte, affiche un message indiquant si le nombre est un entier et qui donne alors le chiffre des dizaines.

Ce que l'on retient :

- les fonctions **floor**, **rand**, et **input**.
- l'affectation des valeurs dans les variables (utilisation du signe =).
- **incrémentation** (a = a + 1) et **initialisation** (a = 1) qui marchent ensemble.
- la notation des opérateurs (surtout == (égale), ~= (différent), **&** (ET) et | (OU)).
- la syntaxe du test : **if – else – end**,