

Devoir 1 : Généralités

- **Chaque réponse ne dépasse pas les 3 lignes** 😊

- 1- C'est quoi la différence entre un **Algorithme** et un **Code** (ex : MatLab).
- 2- C'est quoi la différence entre la boucle **While** (tant que) et **For** (Pour). **YT**

Solutions :

1- La différence entre Algorithme et un Code :

C'est que un algorithme est une série d'étapes pour résoudre : un problème, une tâche ou calculer en langage naturel (Ar, Fr, En), de l'autre côté un code fait la même chose mais en utilisant un langage que la machine peut l'exécuter (langage de haut niveau).

2- La différence entre les boucles **While** (tant que) et **For** (Pour) :

La boucle **While** n'est utilisée que lorsque le nombre d'itérations n'est pas exactement connu. Par contre, lorsque le nombre d'itérations est déjà connu, on utilise la boucle **For**.

Devoir 2: Application I

Ce devoir est composé de trois parties :

Partie I : Définir et explique en détail selon votre connaissance et selon ce que nous avons vu c'est quoi la structure suivante :

IF/SI If...ELSE/ SI...SINON

Partie II : Donne un exemple de votre choix de la structure **IF/SI** et **If...ELSE/ SI...SINON**, et écrit le sous forme d'un **Algorithme en expliquant chaque ligne**. **(Les exemples utilisés dans le Cours/ TP ne sont pas acceptés)**.

Partie III : Transforme l'algorithme en un code MatLab, en expliquant chaque ligne par des commentaires et de même pour le résultat d'exécution.

Voici le lien pour faire l'exécution en ligne : <https://bit.ly/3hrTRZI>



Solutions :

Partie I :

1- La condition : IF/SI

- *SI <condition / expression Booléenne > (VRAI/ FAUX ?)*
ALORS < Instruction ou Bloc d'instruction>
FINSI
- La séquence suivant le mot-clé **ALORS** est exécutée uniquement si l'**Expression Booléenne** renvoie **VRAI**

2- La condition : IF...ELSE/ SI...SINON

- *SI <condition / expression Booléenne > (Vrai/ Faux ?)*
ALORS < Instruction ou Bloc d'instruction>
SINON < Instruction _SINON>
FINSI
- Si le mot-clé **SINON** est atteint, la séquence d'instructions qui le suit est exécutée. Cela se produit lorsqu'aucun des tests conditionnels précédents n'a renvoyé un **VRAI**.

Partie II : Dans cette partie on va présenter deux type d'exemples un pour le chaines de caractères et un pour les nombres.

```

Algorithmme Teste1    % Nom de l'algorithme
Var n : entier        % déclaration de type
Début                 % début de l'algo
Ecrire (' Entrez la valeur de n : ', n) ;
% instructions d'entrées
Lire n ; % et sorties
Si (n > 10) alors % la condition
Ecrire (' n est supérieur de 10') % Action
% ou instruction exécuté si la condition
%est vrai
FinSi % la fin de la condition Si
Fin % Fin d'algo
  
```

```

Algorithmme Teste2    % Nom de l'algorithme
Var s : chaine de caractères
% déclaration de type
Début                 % début de l'algo
Ecrire ('Connaissez-vous Star Wars ? ', 's') ;
% instructions d'entrées
Lire s ; % et sorties
Si (s == ' Oui') alors % la condition
Ecrire ('Que la Force soit avec vous') %Action ou
instruction exécuté si la condition est vrai
SiNon % le cas contraire de la condition
Ecrire (' Bye Bye...') %Action ou %instruction
exécuté si la condition est
%Fausse
FinSi % la fin de la condition Si/Sinon
Fin % Fin d'algo
  
```

Partie III : Code MatLab avec exécution (<https://bit.ly/3hrTRZ1>)

3

% Algorithme Teste1

```
n = input('Entrez la valeur de n : '); % introduire la valeur
                                   %de n operation E/S

if n > 10    % La condition
disp ('n est supérieur de 10') % action de IF
end % fin de la condition if
```

```
1 % Algorithme Teste1
2 n = input('Entrez la valeur de n : '); % introduire la valeur
3                                   %de n operation E/S
4 if n > 10    % La condition
5 disp ('n est supérieur de 10') % action de IF
6 end % fin de la condition if
```

Execute Mode, Version, Inputs & Arguments

GNU 5.1.0

Interactive

CommandLine Arguments

Execute

...

Result

executed in 5.438 sec(s)

Entrez la valeur de n : 11
n est supérieur de 10

%Algorithme Teste2

```

s = input('Connaissez-vous Star Wars ? : ', 's'); % introduire la valeur
                                     %de s operation E/S: comme une chaine
                                     % chaine de caractères

if s == 'oui' % La condition
disp ('Que la Force soit avec vous') % action de IF

else % le cas contraire de If c.à.d "Non"
disp ('Bye Bye...') % action de Else
end % fin de la condition if/else

```

```

1 %Algorithme Teste2
2 s = input('Connaissez-vous Star Wars ? : ', 's'); % introduire la valeur
3                                     %de s operation E/S: comme une chaine
4                                     % chaine de caractères
5 if s == 'oui' % La condition
6 disp ('Que la Force soit avec vous') % action de IF
7
8 else % le cas contraire de If c.à.d "Non"
9 disp ('Bye Bye...') % action de Else
10 end % fin de la condition if/else

```

Execute Mode, Version, Inputs & Arguments

GNU 5.1.0

Interactive

CommandLine Argumenter

Execute

Result

executed in 4.526 sec(s)

```

Connaissez-vous Star Wars ? : oui
Que la Force soit avec vous

```

Execute Mode, Version, Inputs & Arguments

GNU 5.1.0

Interactive

Con

Execute

Result

executed in 18.032 sec(s)

```

Connaissez-vous Star Wars ? : non
Bye Bye...

```

Devoir 3 : Application II

Ce devoir est composé de trois parties :

Partie I : a- Explique chaque ligne de cet algorithme en détail

```
Algorithme Type_Nombre
Var n Entier
Début
Ecrire "Entrez un nombre : "
Lire n
Si n > 0 Alors
Ecrire "Ce nombre est positif"
Sinon
Ecrire "Ce nombre est négatif"
Finsi
Fin
```

b- Ecrire cet algorithme sous forme d'un code MatLab (avec exécution)

Partie II : Explique le résultat de ce code en détail

```
% Exemple de boucle While
N= 6;
while N > 5
disp ('N est nombre entier')
end
```

Partie III :

- a-** C'est quoi le point commun entre la boucle **While (Tant que)** et la condition **IF (SI)**
- b-** Explique le **fonctionnement** de la boucle **For (Pour)**
- c-** C'est quoi l'utilité de **compteur** de la boucle **For (Pour)**

Solutions :

Partie I : a- Explique chaque ligne de cet algorithme en détail

```
Algorithme Type_Nombre      % Nom de l'algo
Var n Entier                %Déclaration de variables
Début                       % Début Algo
Ecrire "Entrez un nombre : " % opérations de
Lire n                      % E/S
Si n > 0 Alors               % la Condition de Si
Ecrire "Ce nombre est positif" % Action Si (Vrai)
Sinon                       % Le cas contraire de Si (Faux)
Ecrire "Ce nombre est négatif" % Action SiNon
Finsi                       % Fin Si/Sinon
Fin                         % Fin Algo
```

b- Ecrire cet algorithme sous forme d'un code MatLab (avec exécution)

```
%Algorithme Type_Nombre
N = input('Entrez un nombre : '); % introduire la valeur de N operation E/S
if N > 0 % La condition
disp ('Ce nombre est positif') % action de IF

else % le cas contraire de If
disp ('Ce nombre est négatif') % action de Else
end % fin de la condition if/else
```

```
1 %Algorithme Type_Nombre
2 N = input('Entrez un nombre : '); % introduire la valeur
3                                     %de N operation E/S
4 if N > 0 % La condition
5 disp('Ce nombre est positif') % action de IF
6
7 else % le cas contraire de If
8 disp('Ce nombre est négatif') % action de Else
9 end % fin de la condition if/else
```

Execute Mode, Version, Inputs & Arguments

GNU 5.1.0



Interactive



Execute

Result

executed in 14.993 sec(s)

```
Entrez un nombre : 5
Ce nombre est positif
```

Execute Mode, Version, Inputs & Arguments

GNU 5.1.0



Interactive



Execute

Result

executed in 7.443 sec(s)

```
Entrez un nombre : -47
Ce nombre est négatif
```

Partie II :

```
% Exemple de boucle While
N= 6;
while N > 5
disp ('N est nombre entier')
end
```

Ce code produise une boucle infinie de **While**. Où le message de l'instruction : **disp ('N est nombre entier')**, s'affiche *répétitivement et infiniment*, car la condition de **While** est toujours **VRAI** ($6 > 5$).

Partie III :

a- C'est quoi le point commun entre la boucle **While** (**Tant que**) et la condition **IF** (**SI**)

✓ **L'expression Booléenne**

b- Explique le **fonctionnement** de la boucle **For** (**Pour**)

✓ **Pour** i = valeur de début : Pas : Valeur de fin
Faire < Instruction ou Bloc d'instruction >
FinPour

✓ Est une structure de contrôle spécifiée pour faire les itérations. **For** permet au code d'être exécuté à plusieurs reprises. Cette boucle est généralement utilisée lorsque le nombre d'itérations est connu d'avance.

c- C'est quoi l'utilité de **compteur** de la boucle **For** (**Pour**)

✓ Le compteur permet au corps de la boucle de savoir quelle itération est en cours d'exécution.