Matlab :

Ce fichier est préparé par **Fayla9o l9owa** d'ENSA Agadir.
∀ error found ∈ doc : contact us on discord.
Let's make ENSA AGADIR great again!

Les structures de contrôle de flux :

Comme en langage C, l'instruction `if` nous permet d'orienter notre code selon la vérification d'une condition :

```
if (condition)
        instruction_1
        instruction_2
        instruction_N
end
```

```
if (condition)
        ensemble dinstructions 1
else
        ensemble dinstructions 2
end
```

Généralement la syntaxe la plus efficace est :

```
if (expression_1)
        Ensemble dinstructions 1
elseif (expression_2)
        Ensemble dinstructions 2
        ....
elseif (expression_n)
        Ensemble dinstructions n
else
        Ensemble dinstructions si toutes les expressions étaient
           fausses
end
```

Manipulation 2 :

Écrire un programme qui calcule et affiche les solutions de l'équation de 2^e degré dans $\mathbb{R}$. (Les coefficients doivent être saisis à partir du clavier).

```
disp('Donner les coefficients de votre polynome')
a=input('donner le coefficient du monome ');
b=input('donner le 2e coeff ');
c=input('donner le 3e coeff ');
d = b^2 - 4*a*c;
if (d > 0)
    disp('Les solutions sont réels')
    x1 = (-b+sqrt(d))/(2*a);
    x2 = (-b-sqrt(d))/(2*a);
    disp(x1, x2);
end
if (d == 0)
    disp('la solutions est double')
    x1= -b/(2*a);
    disp(x1);
else
    disp('les solutions sont complexes')
end
```

Exemple : Écrire une fonction qui lit une matrice carré A et donne son inverse A^{-1}

```
function I=inverse(A)
I=0;
s=size(A);
if s(1)==s(2)
        d=det(A);
        if d ~= 0
                I=A^(-1);
        else
                disp('La matrice ne peut pas être inversée.!');
        end
else
        disp('La matrice ne peut pas être inversée.!');
end
```

Pour l'instruction `switch` on utilise la syntaxe suivante

```
switch (expression)
        case valeur_1
                Groupe dinstructions 1
        case valeur_2
                Groupe dinstructions 2
        case valeur_n
                Groupe dinstructions n
        otherwise
                Groupe dinstructions si tous les case ont échoué
end
```

L'instruction `switch` exécute des groupes d'instructions selon la valeur d'une variable ou d'une expression. Chaque groupe est associé a une clause case qui définie si ce groupe doit être exécuté ou pas selon l'égalité de la valeur de ce case avec le résultat d'évaluation de l'expression de `switch`. Si tous les case n'ont pas étés acceptés, il est possible d'ajouter une clause `otherwise` qui sera exécutée seulement si aucun case n'est exécuté.

Exemple :

```
n = input('Entrer un nombre: ');

switch n
    case -1
        disp('négative')
    case 0
        disp('zéro')
    case 1
        disp('positive')
    otherwise
        disp('autre')
end
```

Pour l'instruction `for`, elle répète l'exécution d'un groupe d'instructions un nombre déterminé de fois. Elle a la forme générale suivante

```
for variable = expression_vecteur
        Groupe d'instructions
end
```

`L'expression vecteur` correspond à la définition d'un vecteur : `début:pas:fin` ou bien `début:fin`.
Étant donnée deux matrices A, B de même dimensions $n \times m$, calculer leur somme :

```
for i = 1:n
        for j = 1:m
                C(i,j) = A(i,j) + B(i,j);
        end
end
```

Pour l'instruction `while` répète l'exécution d'un groupe d'instructions un nombre indéterminé de fois selon la valeur d'une condition logique. Elle a la forme générale suivante :

```
while (condition)
        Ensemble d'instructions
end
```

Exemple : Écrire un programme qui lit un nombre entré et l'affiche tant qu'il est positif :

```
a=input('Entrer un nombre');
while (a<0)
        disp('Essayer une autre fois');
        a=input('Entrer un nombre');
end
disp('Votre nombre est positif');
disp('a');
```