Objective :

- Learn about comparaison and logical operator
- Working with condision and if statemet
- Working with Loops (for,while)
- Creating functions

Comparison and logical operator

Comparison operator:

We can compare 2 variables using 1 of the following operator

- > mean greater then a > b
- < mean less then a < b
- == mean equal a == b
- >= mean greate or equal a >= b
- <= mean less or equal a <= b
- ~= mean not equal a ~= b

result of those operator is logical variabel 0 or 1, 0 represent false 1 represent true

```
a = 5;
b = 6;
disp("Is a greater then b :");
disp(a > b);
disp("Is a less then b :");
disp(a < b);
disp("Is a greater or equal b :");
disp(a >= b);
disp("Is a less or equal b :");
disp(a <= b);
disp("Is a equal b :");
disp(a == b);
disp("Is a different then b (not equal) :");
disp(a ~= b);
```

Logical operator

we use logical operator to combine many condition together in matlab we have 3 ogical operator :

- **&&:** mean condition1 and condition2 it give us 1 only if both of condition1 and condition2 are 1
- **||:** mean condition1 or condition2 it give us 1 if one or both of condition1 and condition2 are 1
- ~: mean not it negate the logical value and reverse it if we have condition1 is 1 not condition1 give us 0

you can see bellow the truth table of those operator

```
a = true;
b = false;
disp(a && b);
disp(a || b);
```

```
disp(~a);
```

Condition and if statement

All the scripts and codes that we wrote so far follow 1 path to excute and it from top to bottom but sometimes we need to add more control to our script and make different path that the script follow depending on some conditions , to do that we use if and else if and else

- we use if to make code excute statement when a condition is true
- we can excute alternative code that run when this condition isn't true by using else
- we use else if to check additional conditions if the first one is false
- after we finish making our condition statement we use end

```
% Example 1: only if, else
n = input("Enter number :")
if n>0
    disp("Positive")
else
    disp("negative or null")
end
% Example 2: if, else if, else
if n>0
    disp("Positive")
elseif n ==0
    disp("null")
else
    disp("negative")
end
```

Loops

We use loops to repeat a statement or block of code more then when time, so instead of keep writing a statement we just write it inside a loop

matlab give us 2 type of loops

For loop

we use this loop in 2 scnario

- when we want to excute statement for all values inside in interval
- when we want to repeat a bloc of code n time

```
% for loop to display all value  in interval
for i = -10:0.1:10
    disp(i*2);
end
```

```
% for loop to repeat a code n time
n = 4;
for j=1:n
```

```
    disp("hello")
end
```

While loop

Generaly we use this loop when we want to excute a statement while a condition is valid , it base on condition more then in interval of value

```
x = 5
while x <= 10
    disp(x)
    x = x + 1
end
```

Remarque

we can fall in infinity loops if we fall on 1 of those error

- Condition in while loop always return 1
- Forget to change the value that made the condition on while loop
- Doing error while making the interval in the for loop

Break and continue

we can use the keywords break and continue to add more control over loop

- break will stop and excute the loop
- continue will skip and jump to next value in the boucle

```
for i=1:10
    if mod(i,2) == 0
        continue % will skip the even number
    elseif i > 5
        break % will quit the loop when i become greater then 5
    end
    disp(i)
end
```

Functions:

Sometimes we have functionality that we want to repeat and use in other scripts ,we can just copy and past this functionality but over time it become annoying , best approche to fix this is creating function file and put this functionality inside it and anytime we want use the functionality we just call the function by it name

Creating function

- Click in new and select function
- Chose ame for the function
- Chose how many value the function should return if it 1 value just use it name else give the values names and put them inside []

- Put the parametre that your function will use inside ()
- Save the file with the same name as your function name

After we create the function and we save it we can use it to our scripts

Using the function:

```
[a,b] = myfunction(4,5);
disp("4 + 5")
disp(a)
disp("4 * 5")
disp(b)
```

Syntext for creating function:

```
function [sum,prod] = myfunction(a,b)
% this function take 2 numeric variable a,b
% and return 2 variable , sum and product of a and b
sum = a + b;
prod = a * b;
end
```

Remarque

In the new version of matlab you can create functions in same file as the code script , to do that just move the function creation bellow your code

We can create documentation and discription of function using comments and anytime we use help myfunction this will display the documentation that we made

Tasks :

Task 1

Using condition statement create script that read 3 value (a,b,c) from the user and solve in R

$$ax^2 + bx + c = 0$$

If the solution is complex number just display "No solution in R"

```

```

Task 2

Create script that read value from user and display all prime number that are less or equal this value

Hint :

prime number is number that accept division by only 1 and itself

```

```

Task 3

Turn the script of the first task into function, the function take 3 parametre, and return 1 value the solution as matrix the equation accept solution, and text say there is no solution if it dont accept solution