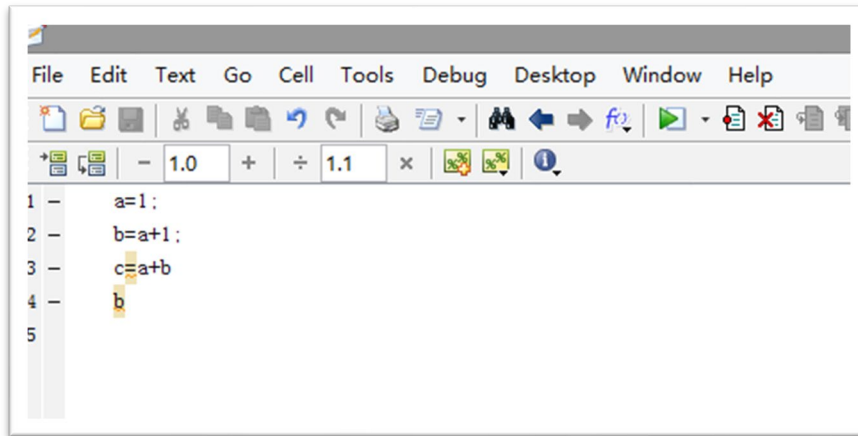
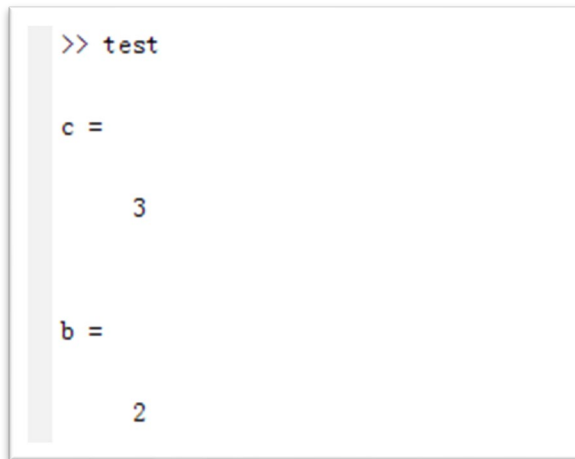


Writing a Script

Writing a Script



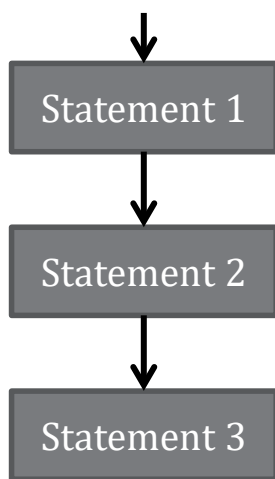
Instead of typing in the **Command Window**, you can write a script in **Editor** and save in a **.m file**, like test.m.



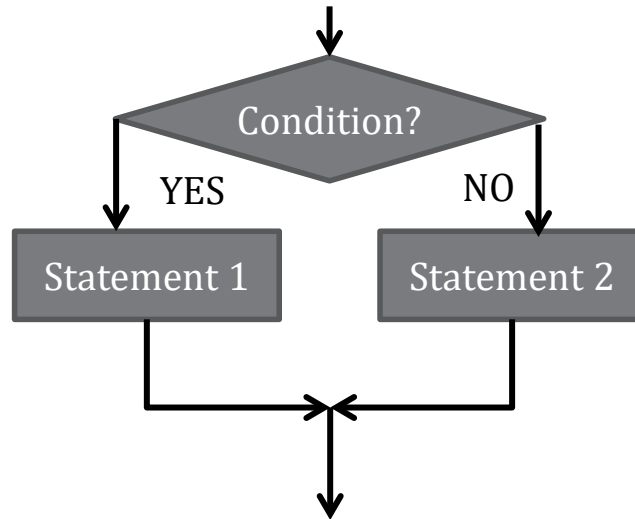
Then, you can run the .m file in **Command Window**. The command in the .m file will be executed one by one (like a batch file).

Flow Control

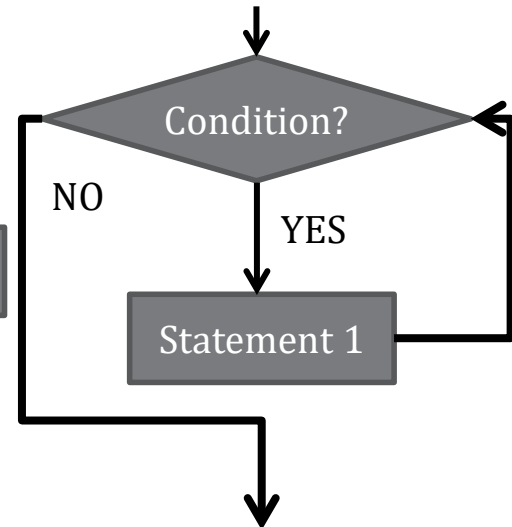
- Similar to almost all programming language, Matlab program has three basic structures
 - Sequence, Selection and Repetition



Sequence



Selection: if



Repetition: for & while

Repetition --- For

- For loop syntax

```
for i=Index_Array  
    Matlab Commands  
end
```

```
for i=1:100  
    Some Matlab Commands;  
end
```

```
for j=1:3:200  
    Some Matlab Commands;  
end
```

```
for m=13:-0.2:-21  
    Some Matlab Commands;  
end
```

```
for k=[0.1 0.3 -13 12 7 -9.3]  
    Some Matlab Commands;  
end
```

- `s = 10;`
- `H = zeros(s);`
- `for c = 1:s`
- `for r = 1:s`
- $H(r,c) = 1/(r+c-1);$
- `end`
- `end`

How many times will the statements in the for loop repeat?

A 65

B 66

C 67

D 68

```
for j=1:3:200
    disp('haha');
end
```

提交

Selection

- If Statement Syntax

```
if (Condition_1)
    Matlab Commands
elseif (Condition_2)
    Matlab Commands
elseif (Condition_3)
    Matlab Commands
else
    Matlab Commands
end
```

```
if ((a>3) & (b==5))
    Some Matlab Commands;
end
```

```
if (a<3)
    Some Matlab Commands;
elseif (b~=5)
    Some Matlab Commands;
end
```

```
if (a<3)
    Some Matlab Commands;
else
    Some Matlab Commands;
end
```

- `nrows = 4;`
- `ncols = 6;`
- `A = ones(nrows,ncols);`
- `for c = 1:ncols`
- `for r = 1:nrows`
- `if r == c`
- `A(r,c) = 2;`
- `elseif abs(r-c) == 1`
- `A(r,c) = -1;`
- `else`
- `A(r,c) = 0;`
- `end`
- `end`
- `end`

- Loop through the matrix and assign each element a new value
- `A(3,5)=?`
- Assign 2 on the main diagonal,
- -1 on the adjacent diagonals, and
- 0 everywhere else.

$A(3,5)=?$

- ☒ A 0
- ☐ B -1
- ☐ C 2
- ☐ D infinity

```

• nrows = 4;
• ncols = 6;
• A = ones(nrows,ncols);
• for c = 1:ncols
•     for r = 1:nrows
•         if r == c
•             A(r,c) = 2;
•         elseif abs(r-c) == 1
•             A(r,c) = -1;
•         else
•             A(r,c) = 0;
•         end
•     end
• end
    
```

提交

Repetition --- While

- While Loop Syntax

```
while (condition)  
    Matlab Commands  
end
```

```
while ((a>3) & (b==5))  
    Some Matlab Commands;  
end
```

- `n = 10;`
 - `f = n;`
 - `while n > 1`
 - `n = n-1;`
 - `f = f*n;`
 - `end`
 - `disp(['n! = ' num2str(f)])`
- `break, return, continue`
 - See MATLAB help

Operators (Logical)

- Non-zero value means true; zero means false
 - 1. == Equal to
 - 2. ~= Not equal to
 - 3. < Strictly smaller
 - 4. > Strictly greater
 - 5. <= Smaller than or equal to
 - 6. >= Greater than equal to
 - 7. & And operator
 - 8. | Or operator
 - 9. ~ Not operator

Script name or function name

- **DO NOT** name your script or function starting with a number.
- **DO NOT** name your script or function as same as the MATLAB build-in function

Writing User Defined Functions

Writing User Defined Functions

- Functions are m-files which can be executed by specifying some inputs and supply some desired outputs.
- The code telling Matlab that an m-file is actually a function is

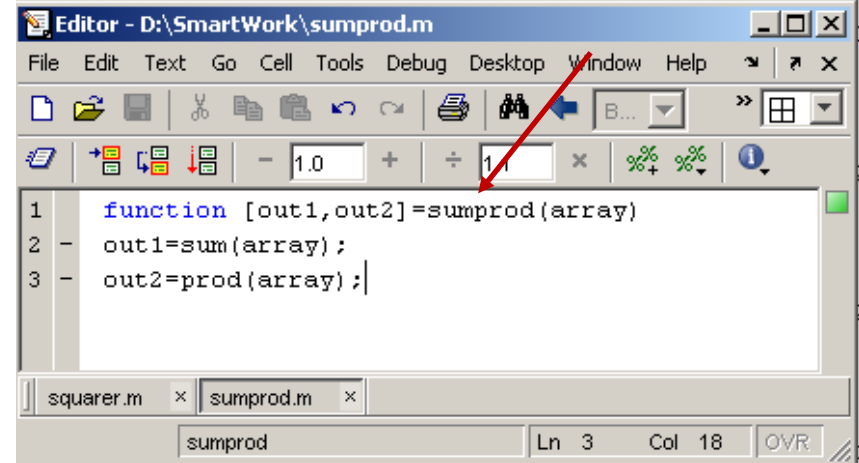
```
function out1=functionname(in1)  
function out1=functionname(in1,in2,in3)  
function [out1,out2]=functionname(in1,in2)
```

- You should write this command **at the beginning of the m-file** and you should save the m-file **with a filename same as the function name.**

Writing User Defined Functions

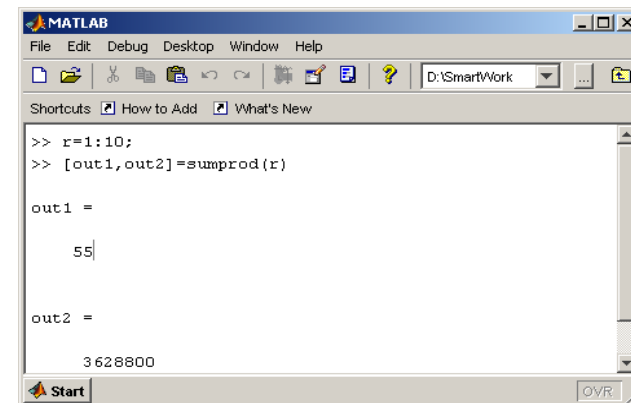
- A function which takes an input array and returns the sum and product of its elements as outputs
 - `function` keyword (required)
 - function name (required)
 - input arguments (optional)
 - output arguments (optional)
- The function `sumprod(.)` can be called from command window or an m-file as

Same Name



The image shows the MATLAB Editor window with the file `D:\SmartWork\sumprod.m` open. The code defines a function `sumprod` that takes an input array and returns its sum and product. A red arrow points from the 'Same Name' text box to the function name `sumprod` in the code.

```
1 function [out1,out2]=sumprod(array)
2 - out1=sum(array);
3 - out2=prod(array);
```



The image shows the MATLAB Command Window with the following commands and output:

```
>> r=1:10;
>> [out1,out2]=sumprod(r)

out1 =

    55

out2 =

 3628800
```


Have a try -1

- Write a function which
 - Generates 2 matrixes of specified size of column and row, and
 - The element value is equal to
 - For matrix 1, the **sum** of column number and row number where it located.
 - For matrix 2, the **product** of
- Eg.

2	3	4		1	2	3
3	4	5	and	2	4	6
4	5	6		3	6	9
- Inputs: size of row, size of column
- Outputs: the desired matrixes

Notes

- “%” is the neglect sign for Matlab (equivalent of “//” in C). Anything after it on the same line is neglected by Matlab compiler.
- Sometimes slowing down the execution is done deliberately for observation purposes. You can use the command “**pause**” for this purpose

```
pause %wait until any key  
pause(3) %wait 3 seconds
```

Have a try -2

- Write a function which
 - Generates 1 matrix of specified size of column and row, and
 - By an input argument, the user can choose the element value is equal to
 - the sum of column number and row number where it located
 - Or, the product of
- Eg.

2	3	4	1	2	3
3	4	5	2	4	6
4	5	6	3	6	9
- Inputs: number of row, number of column, an argument of choice
 - You need `if` statement
- Outputs: the desired matrix

Try to Avoid Loops

- Rather than loops, try to use matrix operators and built-in functions.
- Example: let $a = [a_1 \ a_2 \ a_3]$ and $b = [b_1 \ b_2 \ b_3]$, write a function to calculate

```
c=[a1+b1 a1+b2 a1+b3;  
   a2+b1 a2+b2 a2+b3;  
   a3+b1 a3+b2 a3+b3]
```

- There are two solutions

```
function c = add1(a,b)  
    c = zeros(3,3);  
    for m=1:3  
        for n=1:3  
            c(m,n) = a(m) + b(n);  
        end;  
    end;
```

```
function c = add2(a,b)  
    c = diag(a) * ones(3) + ones(3) * diag(b);
```