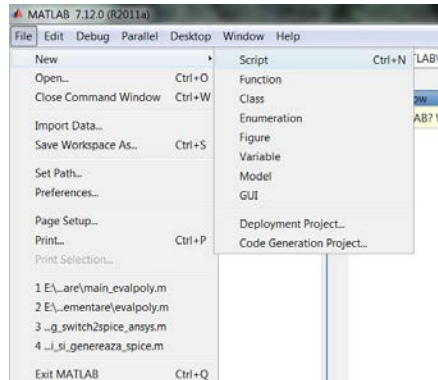


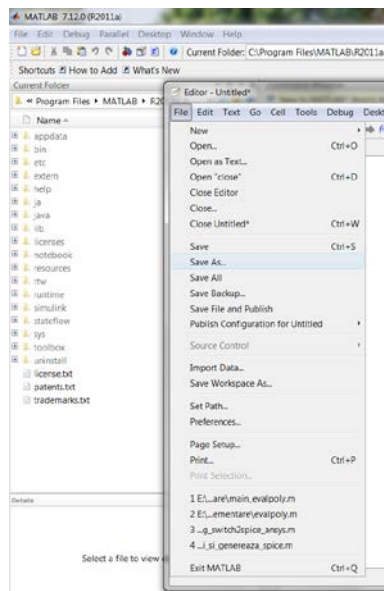
Tutorial de Utilizare și Sintaxă MATLAB

1) Pregătirea pentru implementare

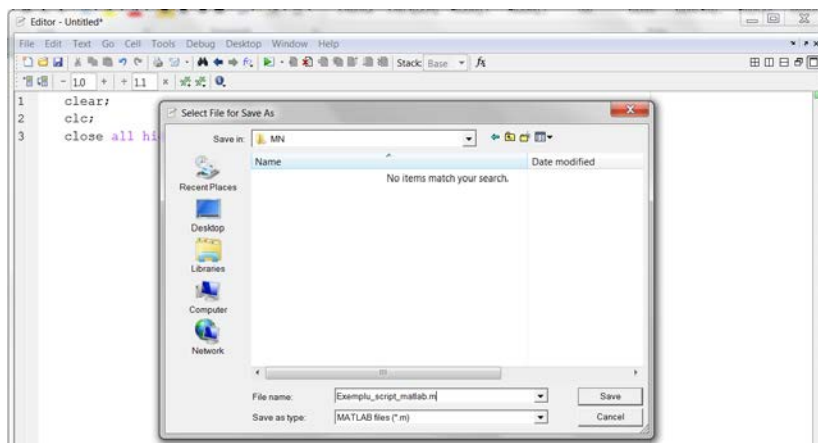
- Deschiderea unui script nou: File -> New -> Script (de la tastatura ctrl+N)



- Pentru a salva scriptul creat, se accesează din meniul scriptului: File -> Save As...



- Se va atribui scriptului nou creat un nume sugestiv:



OBSERVAȚIE: Numele scriptului nu trebuie să conțină spații sau caractere speciale și nu trebuie să înceapă cu o cifră, dacă doriți să delimitați cuvintele se recomandă folosirea: `_,_'`.

2) Atribuirea și folosirea variabilelor:

- Pentru a atribui o valoare unei variabile/constante, se folosește sintaxa:

nume_variabila = valoare

Exemplu:

a = 3; % variabilei a i se atribuie valoarea 3

b = 5; % variabilei b i se atribuie valoarea 5

OBSERVAȚIE: Instrucțiunile se încheie cu punct și virgulă ;

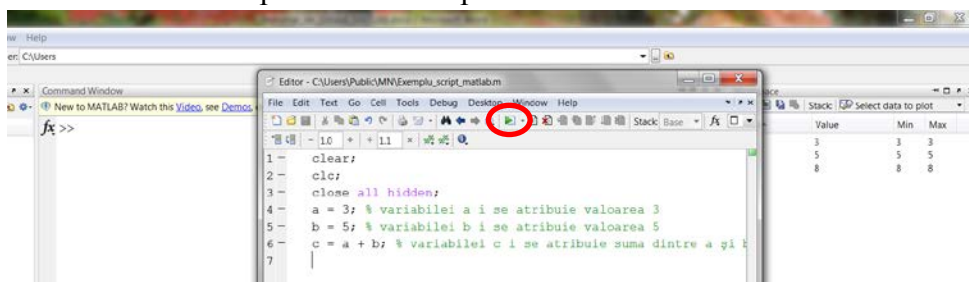
OBSERVAȚIE: Începutul unui comentariu se marchează cu simbolul procent -> %

- După ce variabilelor le-a fost atribuită o valoare, acestea se pot folosi în continuare în calcule.

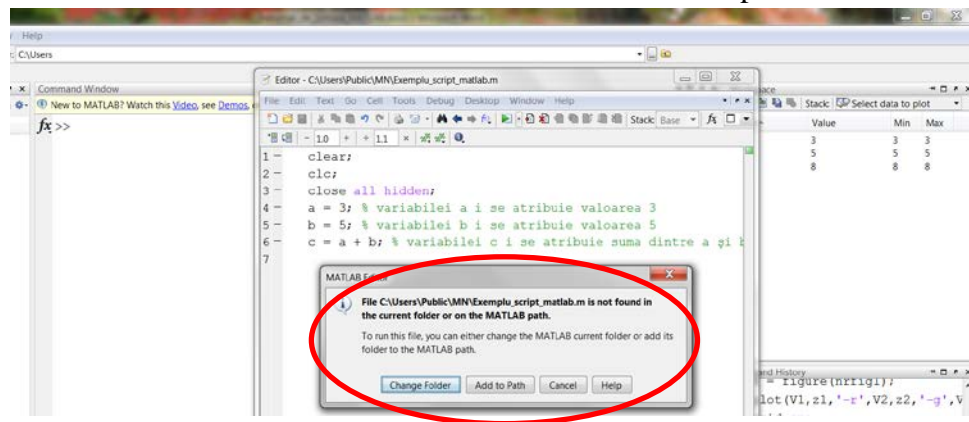
Exemplu:

c = a + b; % variabilei c i se atribuie suma dintre a și b

- Pentru a executa scriptul se dă click pe butonul run aflat în bara cu unelte din script.



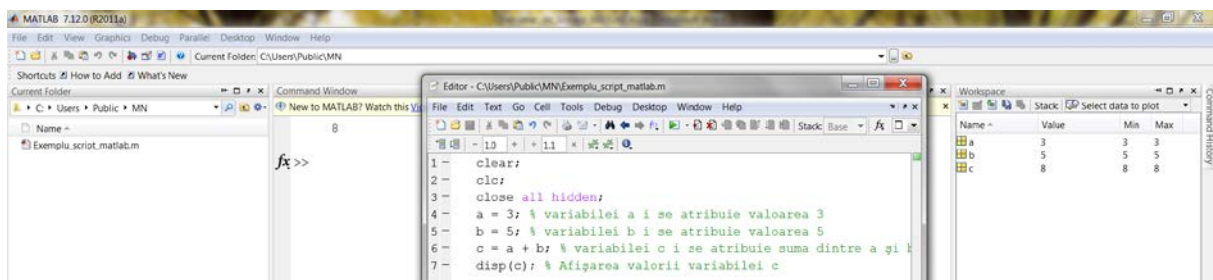
- În cazul în care nu vă aflați în directorul în care a fost salvat scriptul, un mesaj de atenționare va fi afișat pe ecran. Dând click pe butonul Change Folder, programul va seta automat calea către directorul în care a fost salvat scriptul.



- Pentru a afișa valorile variabilelor sau rezultatul unui calcul folosiți comanda *Disp()*. Acestea vor fi afișate în fereastra de comanda (Command Window).

Exemplu:

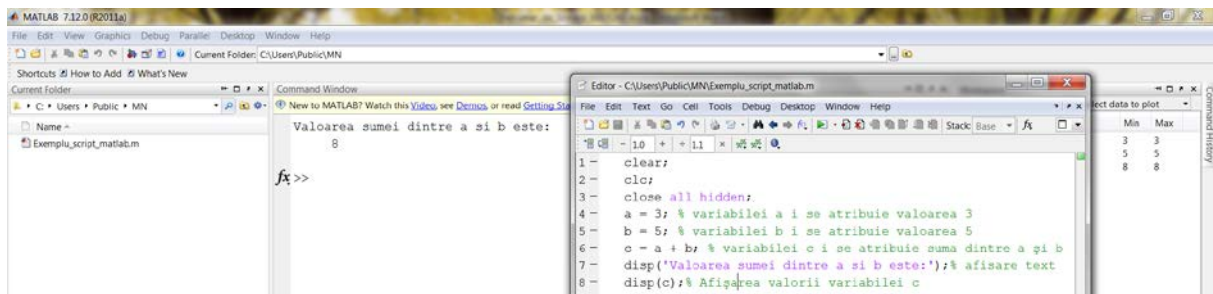
disp(c); % Afișarea valorii variabilei c



- Folosind comanda *disp()* puteți afișa și text, acesta trebuie delimitat de apostrof.

Exemplu:

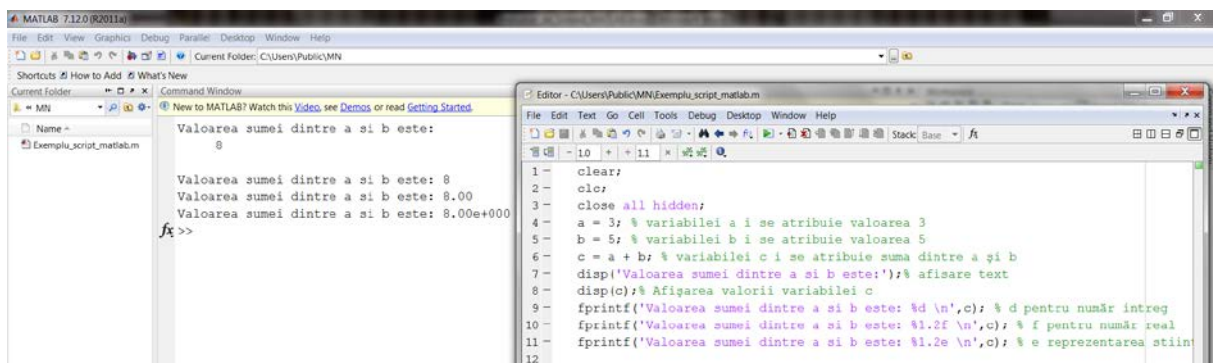
```
disp('Valoarea sumei dintre a si b este:'); % afisare text
disp(c); % Afisarea valorii variabilei c
```



- O comanda de afișa mai avansată este comanda *fprintf()*.

Exemplu:

```
fprintf('Valoarea sumei dintre a si b este: %d \n',c); % d pentru numar intreg
fprintf('Valoarea sumei dintre a si b este: %1.2f \n',c); % f pentru numar real
fprintf('Valoarea sumei dintre a si b este: %1.2e \n',c); % e reprezentarea stiintifica
```

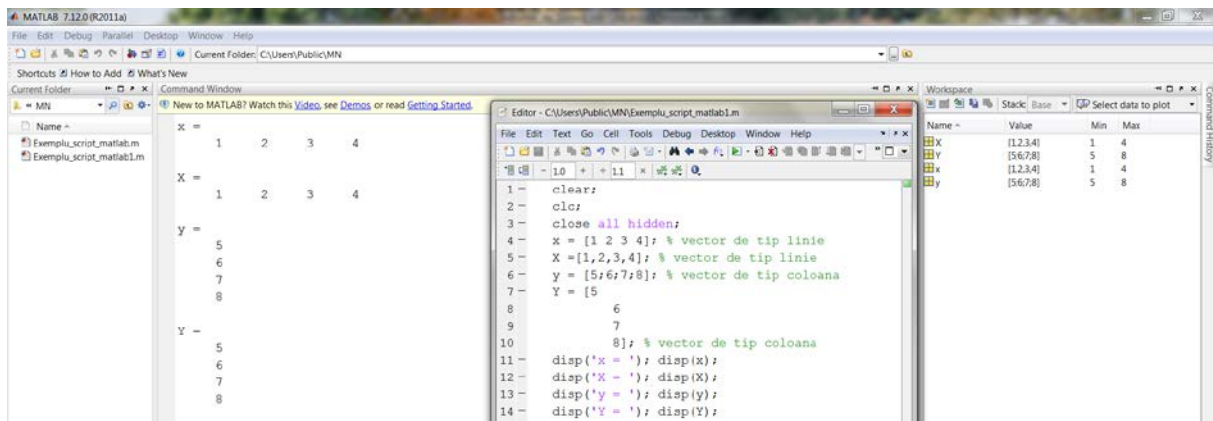


3) Atribuirea și folosirea vectorilor și a matricelor:

- Un vector este structură de date unidimensională cu elemente de acelaș tip. Acesta poate să fie de tip linie (o linie și n coloane) sau de tip coloană (o colonă și n linii).
- În momentul în care se atribuie valori unui vector, acestea trebuie scrise între paranteze pătrate. Șirul de valori trebuie separat folosind spațiu sau virgulă pentru vectorul de tip linie și punct și virgulă sau enter pentru cel de tip coloană.

Exemplu:

```
x = [1 2 3 4]; % vector de tip linie
X = [1,2,3,4]; % vector de tip linie
y = [5;6;7;8]; % vector de tip coloana
Y = [5
6
7
8]; % vector de tip coloana
```



OBSERVAȚIE: MATLAB face distincție între literă mare și literă mică.

- O matrice este o structură de date bidimensională cu elemente de același tip. Aceasta conține n linii cu m coloane.
- În momentul în care se atribuie valori unui matrice, acestea trebuie scrise între paranteze pătrate. Elementele de pe linii trebuie separate folosind spațiu sau virgulă și liniile se separă folosind punct și virgulă sau enter.

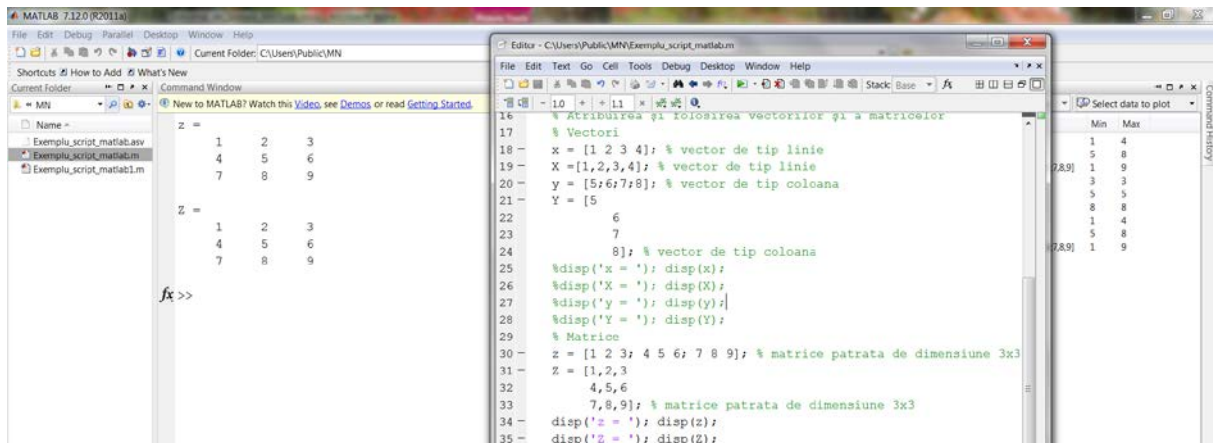
Exemplu:

$z = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9];$ % matrice patrata de dimensiune 3x3

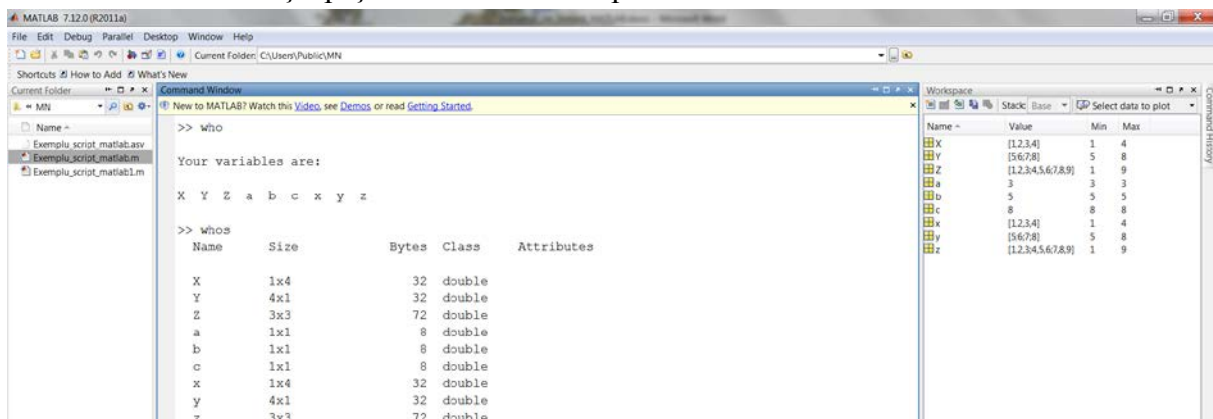
$Z = [1,2,3$

$4,5,6$

$7,8,9];$ % matrice patrata de dimensiune 3x3



- Comanda **who** afișează variabilele folosite. Comanda **whos** afișează variabilele folosite, dimensiunea și spațiul de memorie ocupat de acestea.



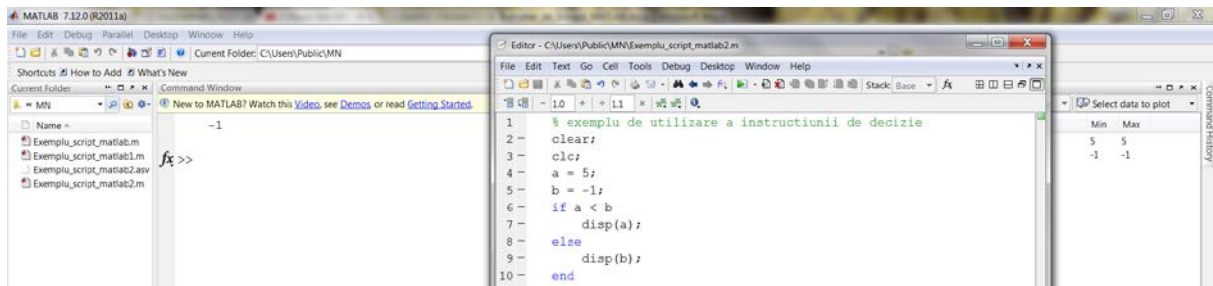
4) Instrucțiunea de decizie *if*

- Sintaxa:

```
if expresie
    instructiune1;
else
    instructiune2;
end
```

Exemplu: compararea valorilor a două variabile și afișarea celei mai mici valori.

```
a = 5;
b = -1;
if a < b
    disp(a);
else
    disp(b);
end
```

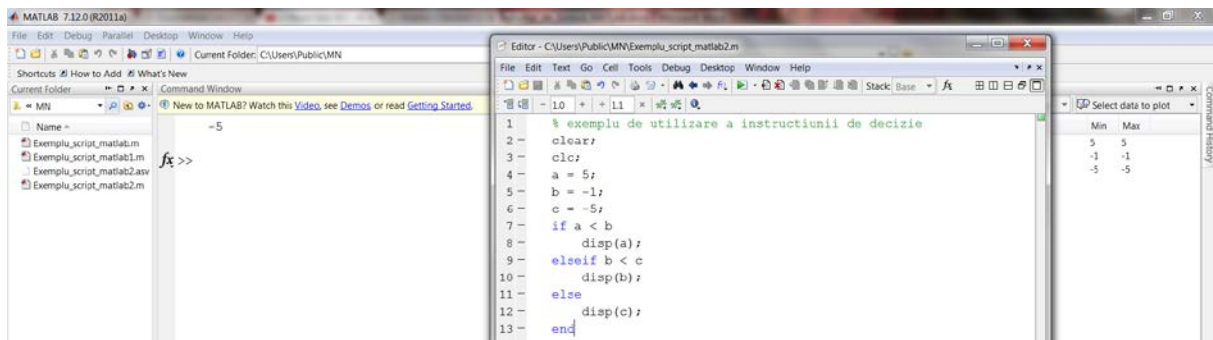


- Dacă este nevoie de imbricarea a mai multor instrucțiuni de decizie sintaxa devine:

```
if expresie 1
    instructiune 1;
elseif expresie 2
    instructiune 2;
elseif expresie 3
    instructiune 3;
    -----
else
    instructiune n;
end
```

Exemplu: compararea valorilor a trei variabile și afișarea celei mai mici valori.

```
a = 5;
b = -1;
c = -5;
if a < b
    disp(a);
elseif b < c
    disp(b);
else
    disp(c);
end
```

5) Instrucțiunea ciclului cu contor *for*

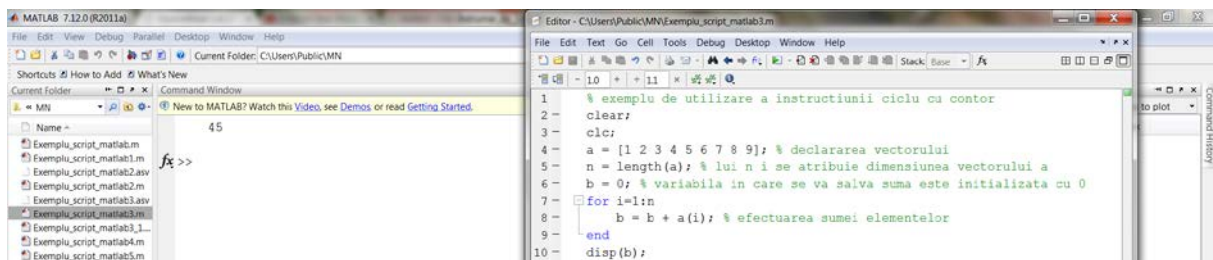
- Sintaxa:

```
for contor = valoare_inicial : pas_incrementare : valoare_finala
    instructiune;
end
```

OBSERVAȚIE: În cazul în care pasul de incrementare este 1, *pas_incrementare* poate să lipsească, declarându-se doar valoarea inițială și valoarea finală.

Exemplul 1: suma elementelor unui vectori de dimensiunea n.

```
a = [1 2 3 4 5 6 7 8 9]; % declararea vectorului
n = length(a); % lui n i se atribuie dimensiunea vectorului a
b = 0; % variabila in care se va salva suma este initializata cu 0
for i=1:n
    b = b + a(i); % efectuarea sumei elementelor
end
disp(b);
```



- Comanda *length* determină numărul de elemente dintr-un vector.

Exemplul 2: suma elementelor de pe pozițiile pare ale unui vector de dimensiunea n.

```
a = [1 2 3 4 5 6 7 8 9]; % declararea vectorului
n = length(a); % lui n i se atribuie dimensiunea vectorului a
b = 0; % variabila in care se va salva suma este initializata cu 0
for i=2:2:n % se porneste de la prima pozitie para
    b = b + a(i); % efectuarea sumei elementelor
    disp(a(i)); % afisarea valorii de pe pozitia i
end
disp(b);
```

```

1 % exemplu de utilizare a instructiunii ciclului cu contor
2 clear;
3 clc;
4 a = [1 2 3 4 5 6 7 8 9]; % declararea vectorului
5 n = length(a); % lui n i se atribuie dimensiunea vectorului a
6 b = 0; % variabila in care se va salva suma este initializata cu 0
7 for i=2:2:n % se porneste de la prima pozitie para
8     b = b + a(i); % efectuarea sumei elementelor
9     disp(a(i)); % afisarea valorii de pe pozitia i
10 end
11 disp(b);

```

Command Window Output:

```

2
4
6
8
20

```

Exemplul 3: Afișarea în ordine inversă a elementelor unui vector.

```

a = [1 2 3 4 5]; % declararea vectorului
n = length(a); % lui n i se atribuie dimensiunea vectorului a
for i=n:-1:1 % se porneste de la valoarea finala care se decrementeaza
    % cu -1 pana se ajunge la valoarea initiala
    disp(a(i)); % afisarea valorii de pe pozitia i
end

```

```

1 % exemplu de utilizare a instructiunii ciclului cu contor
2 clear;
3 clc;
4 a = [1 2 3 4 5]; % declararea vectorului
5 n = length(a); % lui n i se atribuie dimensiunea vectorului a
6 for i=n:-1:1 % se porneste de la valoarea finala care se decrementeaza
7     % cu -1 pana se ajunge la valoarea initiala
8     disp(a(i)); % afisarea valorii de pe pozitia i
9 end

```

Command Window Output:

```

5
4
3
2
1

```

6) Instrucțiunea ciclului cu test inițial **while**

- Sintaxa:
while *expresie*
 instructiune;
end

Exemplul 1: Determinarea celui mai mare număr care poate fi reprezentat pe calculator.

```

b=1; % variabila folosita pentru calculul celui mai mare numar
k=0; % se foloseste un contor care sa numere puterea lui 10
while b~=inf % inf reprezinta infinit, cat timp b este diferit de infinit repeta
    a=b; % memorarea valorii de la pasul anterior
    k=k+1;
    b=b*10;
end
disp(k);
disp(a);

```

```

1 % exemplu de utilizare a instructiunii ciclului cu test
2 clear;
3 clc;
4 b=1; % variabila folosita pentru calculul celui mai mare numar
5 k=0; % se foloseste un contor care sa numere puterea lui 10
6 while b~=inf % inf reprezinta infinit
7     a=b; % memorarea valorii de la pasul anterior
8     k=k+1;
9     b=b*10;
10 end
11 disp(k);
12 disp(a);

```

Command Window Output:

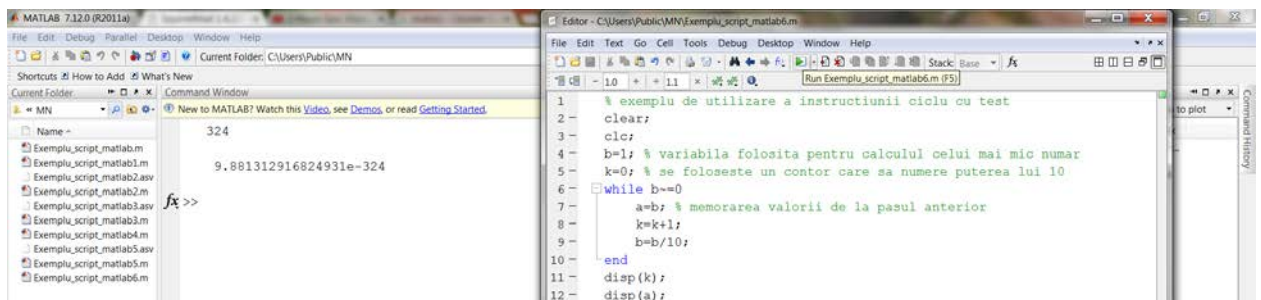
```

309
9.999999999999999e+307
fx >>

```

Exemplul 2: Determinarea celui mai mic număr pozitiv care poate fi reprezentat pe calculator.

```
b=1; % variabila folosita pentru calculul celui mai mic numar
k=0; % se foloseste un contor care sa numere puterea lui 10
while b~=0 % cat timp b este diferit de 0 repeta
    a=b; % memorarea valorii de la pasul anterior
    k=k+1;
    b=b/10;
end
disp(k);
disp(a);
```



The screenshot shows the MATLAB environment. The script editor on the right contains the code from the previous block. The command window on the left shows the output of the script: 324 and 9.881312916824931e-324.

OBSERVAȚIE: În MATLAB nu este implementat ciclul cu test final.

7) Pentru aprofundarea sintaxei MATLAB, vă recomandăm parcurgerea următoarelor referințe:

1. G.Ciuprina, Algoritmi numerici prin exerciții și implementări în Matlab, Editura MatrixROM, 2013, ISBN 978-606-25-0009-2,
http://lmn.pub.ro/~gabriela/books/AlgNrExMatlab_MatrixRom2013.pdf
2. Tutorials Point, MATLAB numerical computing, 2013,
https://web.njit.edu/~yyoung/M222Spring2016/matlab_tutorial.pdf
3. David Houcque, Introduction to Matlab for Engineering Students, Northwestern University 2005,
<https://www.mccormick.northwestern.edu/documents/students/undergraduate/introducti-on-to-matlab.pdf>
4. Holly Moore, MATLAB® for Engineers, www.arifsari.net/downloads/MATLAB.pdf
5. Introduction to MATLAB, www.youtube.com/watch?v=7bnVx34yQf4
6. Introduction to MATLAB for Engineers, www.youtube.com/watch?v=St0eA7bzLQU
7. MATLAB Tutorial for Beginners, www.youtube.com/watch?v=bdw31FBdyk
8. An introduction to coding in MATLAB, www.youtube.com/watch?v=GEawbB8nZso
9. Programming with MATLAB, www.youtube.com/watch?v=-JQPGKMHmPY