Tema lab. 6

NICOLETA RADU

# Exercitiul 1

Să se scrie o funcţie în Matlab, care are ca date de intrare o matrice A şi returnează o valoare x definită astfel:

• x=1, dacă există în matricea A cel puțin un element mai mare decât o valoare citită de la tastatură;

• x=2, dacă există în matricea A cel puțin un element mai mic decât o valoare obținută aleatoriu din domeniul (5,6);

• x=0, altfel. Funcţia va semnala eroare dacă matricea nu conține numere finite.

function [x] = exe1(A)

%UNTITLED Summary of this function goes here

% Detailed explanation goes here

userValue = input('Dati o valoare:')

[l,c] = size(A)

randValue = round(5 + (6-5)\*rand(1))

x = 0;

if isfinite(A) == 0

error = 'Matricea contine numere finite'

end

for lin = 1:l

for col = 1:c

if A(lin,col) > userValue

x = 1;

break

elseif A(lin,col) < randValue

x = 2;

break

else

x = 0;

break

end

end

end

# Exercitiul 2

Să se scrie o funcţie Matlab care primeşte ca argumente trei valori a, b, p şi care generează vectorul cu pas liniar v=a:p:b. Funcţia returnează vectorul generat, lungimea sa, precum şi suma elementelor sale. Să se testeze funcţia creată pentru următoarele triplete de valori: (0,25,5), (2,19,3), (5,-3,-2), (5,-2,0), (2,19,-1).

function [vectorFormat,marimeVector,sumaElem] = exe2(a,b,p)

%UNTITLED3 Summary of this function goes here

% Detailed explanation goes here

vectorFormat = (a:p:b);

marimeVector = length(vectorFormat)

sumaElem = 0;

for index = 1:marimeVector

sumaElem = sumaElem + vectorFormat(index);

end

sumaElem

end

a = exe2(0,25,5)

b = exe2(2,19,3)

c = exe2(5,-3,-2)

d = exe2(5,-2,0)

e = exe2(2,19,-1)

# Exercitiul 3

Pentru o matrice pătratică transmisă ca parametru unei funcţii Matlab se cere să se scrie o secvenţă de instrucţiuni prin intermediul căreia: să se specifice dacă matricea este inversabilă (adică, det(A)≠0); în cazul în care matricea este inversabilă, să se afişeze inversa ei, iar în cazul în care matricea nu este inversabilă, să se afişeze rangul ei. Construiți un fișier script de test pentru funcția de mai sus și testați-o pentru fiecare din matricele A=[1 0; -1 1]; B=[1 1; 2 2]; C=[1 1 2; 2 2 3];

A=[1 0; -1 1];

B=[1 1; 2 2];

C=[1 1 2; 2 2 3];

a = exe3(A)

b = exe3(B)

c = exe3(C)

function [inversa] = exe3(myMatrix)

%UNTITLED5 Summary of this function goes here

% Detailed explanation goes here

inversa = 0;

if det(myMatrix) ~= 0

inversa = inv(myMatrix);

else

rang = rank(myMatrix)

end

>> exer\_4

a =

1 0

1 1

rang =

1

b =

0

Error using det

Matrix must be square.

Error in exe3 (line 7)

if det(myMatrix) ~= 0

Error in exer\_4 (line 7)

c = exe3(C)

>>

# Exercitiul 4

Se consideră f R R : → definită prin: 2 1 , 1 2 1 ( ) 0, 1 ln( 2 2), 1 pentru x x f x pentru x x x pentru x ⎧ < ⎪ − ⎪ = = ⎨ ⎪ − + > ⎪ ⎩ Să se definească un fișier de tip funcție pentru implementarea funcției f(x). Să se verifice pentru cazurile x=2, x=1, x=-2 și pentru o mulțime de 10 numere echidistante din domeniul [-5,5].

x = 2;

rez1 = exe4(x)

x = 1;

rez2 = exe4(x)

x = -2;

rez3 = exe4(x)

vect = (-5:2:5);

marimeVect = length(vect);

for index = 1:marimeVect

rezVector = exe4(vect(index))

end

function [rezultat] = exe4(x)

%UNTITLED6 Summary of this function goes here

% Detailed explanation goes here

if x < 1

rezultat = 1 / (2 \* x - 1);

elseif x == 1

rezultat = 0;

elseif x > 1

rezultat = log(x^2 - 2 \* x + 2);

else

error = 'Valoarea introdusa este ilegala'

end

# Exercitiul 5

Se consideră mulțimea primelor n numere naturale: M={1,2,3,…,n}. Să se scrie o funcție pentru calculul produsului elementelor mulțimii M, folosind o instrucțiune repetitivă: 1 1 2 ... n i P n i = = ⋅ ⋅ ⋅ =∏ . Să se verifice funcția pentru n=5, n=10 și n=15, apoi să se verifice rezultatele obținute folosind funcția MATLAB predefinită prod. Pentru verificare se va folosi un fișier script.

**Pentru n = 5**

n = 5;

M = 1:n;

rezultat\_1 = exe5(M);

rezultat\_2 = prod(M);

function [produs] = exe5(yourVector)

%UNTITLED5 Summary of this function goes here

% Detailed explanation goes here

n = length(yourVector);

produs = 1;

for index = 1:n

produs = produs \* yourVector(index);

end

**pentru n=10**

>> tema6

rezultat\_1 =

3628800

rezultat\_2 =

3628800

>>

**Pentru n=15**

>> tema6

rezultat\_1 =

1.3077e+12

rezultat\_2 =

1.3077e+12

>>

# Exercitiul 6

Să se construiască un fișier funcție care primește ca argumente de intrare două numere naturale n și m, unde n>m și un număr x din mulțimea {1,2,3}.

• Dacă x = 1 se va calcula permutări de n: P n = ! (unde n n n ! 1 2 ... ( 1) = ⋅ ⋅ ⋅ − ⋅ )

• Dacă x = 2 se va calcula aranjamente de n luate câte m: ! ( )! m n n A n m = −

• Dacă x = 3 se va calcula combinări de n luate câte m: ! !( )! m n n C m n m = −

• Dacă x este un număr din afara elementelor mulțimii, se va calcula implicit permutări.

• Dacă m, n nu respectă cerința că n>m, se vor inversa numerele între ele (numărul mai mare va fi atribuit lui n, iar numărul mai mic lui m).

• Dacă m==n, se va calcula implicit permutări.

Pentru calculul factorialului numărului n se va construi o funcție separată numită factorial\_n(). Folosind un fișier script să se verifice funcția de mai sus pentru următoarele triplete de valori: (n=8,m=5,x=1), (n=8,m=5,x=2), (n=8,m=5,x=3), (n=5,m=5,x=3), (n=5,m=8,x=2). Să se verifice apoi rezultatele folosind funcția MATLAB predefinită factorial.

function [p,a,c] = exe6(n,m,x)

%UNTITLED Summary of this function goes here

% Detailed explanation goes here

p = 1;

if n > m

if x == 1

p = factorial\_n(n);

elseif x == 2

a = factorial\_n(n) / (factorial\_n(n-m));

elseif x == 3

c = factorial\_n(n) / (factorial\_n(m) \* factorial\_n(n-m));

else

message = 'X inafara multimii'

p = factorial\_n(n);

end

elseif n == m

message = 'n == m'

p = factorial\_n(n);

else

aux = n;

n = m;

m = aux;

if x == 1

p = factorial\_n(n);

elseif x == 2

a = factorial\_n(n) / (factorial\_n(n-m));

elseif x == 3

c = factorial\_n(n) / (factorial\_n(m) \* factorial\_n(n-m));

else

message = 'X inafara multimii'

p = factorial\_n(n);

end

end

function [prod] = factorial\_n(n)

%UNTITLED3 Summary of this function goes here

% Details./..

prod = 1;

for ind = 1:n

prod = prod \* ind;

end

n=8;

m=5;

x=1;

rez\_1 = exe6(n,m,x) % permutari

n=8;

m=5;

x=2;

rez\_2 = exe6(n,m,x) % aranjamente

n=8;

m=5;

x=3;

rez\_3 = exe6(n,m,x) % combinari

n=5;

m=5;

x=3;

rez\_4 = exe6(n,m,x) % m > n

n=5;

m=8;

x=2;

rez\_5 = exe6(n,m,x)

>> tema6

rez\_1 =

40320

rez\_2 =

1

rez\_3 =

1

message =

'n == m'

rez\_4 =

120

rez\_5 =

1

>>

# Exercitiul 7

Construiți o funcție care calculează media geometrică a elementelor unui vector: 1 1 2 1 ... n n n g n i i m a a a a = ⎛ ⎞ = ⋅ ⋅ ⋅ = ⎜ ⎟ ⎝ ⎠ ∏ . Pentru calculul produsului se va folosi instrucțiunea while. Folosind un fișier script să se verifice funcția de mai sus pentru:

• Un vector cu 10 numere întregi din domeniul [5,10);

• Un vector cu 6 numere echidistante în domeniul [-2,6];

• Un vector cu elemente din domeniul [5,10] și pas 2.5.

a = 5:0.55:10

b = -2:1.5:6

c = 5:2.5:10

rez1 = mg(a)

rez2 = mg(b)

rez3 = mg(c)

function [mediaG] = mg(yourVector)

%UNTITLED6 Summary of this function goes here

% Detailed explanation goes here

mSize = length(yourVector);

mediaG = 1;

prod = 1;

while mSize > 0

prod = prod \* yourVector(index);

mSize = mSize – 1;

end

mediaG = sqrt(prod);

end

>> tema6

a =

5.0000 5.5500 6.1000 6.6500 7.2000 7.7500 8.3000 8.8500 9.4000 9.9500

b =

-2.0000 -0.5000 1.0000 2.5000 4.0000 5.5000

c =

5.0000 7.5000 10.0000

rez1 =

2.0774e+04

rez2 =

7.4162

rez3 =

19.3649

>>

# Exercitiul 8

Construiți un fișier funcție pentru a înlocui într-o matrice, dată ca argument de intrare, toate elementele pare cu inf folosind două moduri de rezolvare (prima prin instrucțiuni de control, a doua metodă prin comenzi Matlab). Funcția va avea ca argumente de intrare matricea și modul de calcul (1 sau 2). Dacă se citește o altă valoare decât 1 sau 2, atunci implicit se va alege prima metodă.

function [newMatrix] = exe8(yourMatrix,option)

%UNTITLED7 Summary of this function goes here

% Detailed explanation goes here

[l,c] = size(yourMatrix);

newMatrix = zeros(l,c);

% prin instructiuni de control

if option == 1

for indexl = 1:l

for indexc = 1:c

if rem(yourMatrix(indexl,indexc),2) == 0

yourMatrix(indexl,indexc) = inf;

end

end

end

elseif option == 2

[row,col] = find(rem(yourMatrix,2) == 0);

yourMatrix(sub2ind(size(yourMatrix),row(:),col(:))) = inf;

else

[row,col] = find(rem(yourMatrix,2) == 0);

yourMatrix(sub2ind(size(yourMatrix),row(:),col(:))) = inf;

end

% prin comenzi matlab

for indexl = 1:l

for indexc = 1:c

newMatrix(indexl,indexc) = yourMatrix(indexl,indexc);

end

end

n = 1

M = randi([1,20],5);

exe8(M,n)

n = 1

M = randi([1,20],5);

exe8(M,n)

>> tema6

n =

1

M =

20 4 7 14 2

11 9 11 10 18

18 15 2 4 12

12 17 3 10 19

4 16 3 3 14

ans =

Inf Inf 7 Inf Inf

11 9 11 Inf Inf

Inf 15 Inf Inf Inf

Inf 17 3 Inf 19

Inf Inf 3 3 Inf

>> tema6

n =

2

ans =

Inf Inf 17 17 Inf

17 13 5 15 13

Inf Inf Inf Inf Inf

Inf 11 19 5 15

1 Inf Inf Inf Inf

>>

# Exercitiul 9

Construiți o funcție care calculează media pătratică a elementelor unui vector: 1 2 2 2 2 1 2 2 1 ... 1 n n p i i a a a m a n n = + + + ⎛ ⎞ = = ⎜ ⎟ ⎝ ⎠ ∑ . Pentru calculul sumei se va folosi instrucțiunea for. Folosind un fișier script să se verifice funcția de mai sus pentru: • Un vector cu 10 numere întregi din domeniul [-5,10); • Un vector cu 5 numere echidistante în domeniul [-2,2]; • Un vector cu elemente din domeniul [5,10] și pas 2.5.

function [Mp] = exe9(yourVector)

%UNTITLED Summary of this function goes here

% Detailed explanation goes here

s = length(yourVector);

suma = 0;

for index = 1:s

suma = suma + (yourVector(index).^2);

end

suma = suma / s;

Mp = sqrt(suma);

end

a = -5:1.55:10;

b = -2:1:2;

c = 5:2.5:10;

rez1 = exe9(a)

rez2 = exe9(b)

rez3 = exe9(c)

>> tema6

rez1 =

4.8704

rez2 =

1.4142

rez3 =

7.7728

>>

# Exercitiul 10

Să se construiască un fișier funcție pentru determinarea numărului de elemente pozitive (P), a numărului de elemente negative (N), respectiv a numărului de elemente egale cu zero (O) ale unui vector, dat ca argument de intrare. Să se verifice funcția pentru a=[-1,2,0,4,-5,-6,0,8, 9,10] și pentru un vector linie de 20 de elemente, numere întregi, generate aleatoriu din domeniul [-5,5). Funcția va avea ca argumente de ieșire valorile lui P, N și O.

function [P,N,O] = exe10(yourVector)

%UNTITLED2 Summary of this function goes here

% Detailed explanation goes here

vectorSize = length(yourVector);

P = 0;

N = 0;

O = 0;

for index = 1:vectorSize

if yourVector(index) == 0

O = O + 1;

elseif yourVector(index) > 0

P = P + 1;

elseif yourVector(index) < 0

N = N + 1;

end

end

P

N

O

end

a=[-1,2,0,4,-5,-6,0,8,9,10]

b= randi([-5,5],1,20)

rez1 = exe10(a)

rez2 = exe10(b)

>> tema6

a =

-1 2 0 4 -5 -6 0 8 9 10

b =

Columns 1 through 16

4 -1 4 -1 3 -1 3 3 -1 -3 3 5 -2 2 -1 4

Columns 17 through 20

3 -4 4 5

P =

5

N =

3

O =

2

rez1 =

5

P =

12

N =

8

O =

0

rez2 =

12

>>

# Exercitiul 12

Să se construiască un fișier funcție pentru calculul sumei elementelor unui vector a care aparțin domeniului b. Funcția va avea 2 argumente de intrare a și b, unde b este un vector linie de 2 elemente, reprezentând domeniul de definiție pentru elementele căutate în vectorul a. Se cere:

• Dacă b nu este citit corect (vector linie de 2 elemente), să se afișeze eroare; • Dacă a este un vector vid, să se genereze un vector de 10 elemente numere întregi din domeniul [5,15);

• Dacă a este o matrice, să se transforme matricea intr-un vector linie; • Dacă numărul de argumente de intrare nu este 2, se afișeze un mesaj de eroare; Să se verifice pentru a=[1,2,3,…,10] și b=[4 8].

x = [0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 1.4, 1.6, 1.8, 2];

s = length(x);

for index = 1:s

fct(x(index))

end

>> tema6

ans =

0

ans =

0.4214

ans =

0.8918

ans =

1.4221

ans =

2.0255

ans =

2.7183

ans =

0.9642

ans =

0.8741

ans =

0.7543

ans =

0.6249

ans =

0.5000

>>

function [rezultat] = fct(x)

%UNTITLED2 Summary of this function goes here

% Detailed explanation goes here

if x <=1

rezultat = exp(x) + x - 1;

else

rezultat = 1 / (x^(x-1));

end

# Exercitiul 13

Să se construiască un fişier de tip funcţie care să genereze o matrice cu m linii şi n coloane având pe coloanele impare caracterul \* şi pe coloanele pare caracterul @. Să se verifice funcția pentru următoarele perechi de valori (2,3), (8,5), (5,10).

function [yourMatrix] = exe13(m,n)

%UNTITLED3 Summary of this function goes here

% Detailed explanation goes here

for indL = 1:m

for indC = 1:n

yourMatrix(indL,indC) = '\*';

end

end

mat2str(yourMatrix);

for indexL = 1:m

for indexC = 1:n

if rem(indexC,2) == 1

yourMatrix(indexL,indexC) = '\*';

else

yourMatrix(indexL,indexC) = '@';

end

end

end

a = 2;

b = 3;

exe13(a,b)

a = 8;

b = 5;

exe13(a,b)

a = 5;

b = 10;

exe13(a,b)

>> tema6

ans =

2×3 char array

'\*@\*'

'\*@\*'

ans =

8×5 char array

'\*@\*@\*'

'\*@\*@\*'

'\*@\*@\*'

'\*@\*@\*'

'\*@\*@\*'

'\*@\*@\*'

'\*@\*@\*'

'\*@\*@\*'

ans =

5×10 char array

'\*@\*@\*@\*@\*@'

'\*@\*@\*@\*@\*@'

'\*@\*@\*@\*@\*@'

'\*@\*@\*@\*@\*@'

'\*@\*@\*@\*@\*@'

>>

# Exercitiul 14

Să se construiască un fişier de tip funcţie care să contorizeze de câte ori apare o valoare val în cadrul unei matrice date, A. Funcția va avea ca argumente de intrare matricea și valoarea.

function [count] = gasesteValoarea(yourMatrix,yourVar)

%UNTITLED4 Summary of this function goes here

% Detailed explanation goes here

[l,c] = size(yourMatrix);

count = 0;

for indL = 1:l

for indC = 1:c

if yourMatrix(indL,indC) == yourVar

count = count + 1;

end

end

end

**SCRIPT**

M = randi([1,15],4)

val = 10;

gasesteValoarea(M,val)

**REZULTAT**

>> tema6

ans =

1

>> tema6

M =

11 10 15 12

12 3 6 4

5 2 9 8

11 8 4 11

ans =

1

>> tema6

M =

14 3 13 3

15 4 4 4

9 13 14 10

3 4 6 8

ans =

1

>> tema6

M =

6 14 6 8

13 5 9 12

9 12 2 15

9 12 1 2

ans =

0

>> tema6

M =

9 3 3 11

8 12 10 12

1 5 4 7

6 8 10 2

ans =

2

>>

# Exercitiul 15

Să se construiască un fișier funcție prin care să se schimbe ordinea coloanelor unei matrice M astfel încât elementele din linia selectată să fie ordonate crescător. Funcția va primi ca argumente de intrare matricea și linia care trebuie ordonată. Se se verifice funcția pentru: M = [1 4 2; 4 5 3; 1 4 5; 2 4 6], lin=2.

**Functia 1**

**Schimbarea coloanelor**

function [newMat] = exe15(yourMAtrix,col\_1,col\_2)

%UNTITLED5 Summary of this function goes here

% Detailed explanation goes here

[l,c] = size(yourMAtrix);

newMat = zeros(l,c);

r = 0;

copyC = c;

% if columns nums are smaller than rows

if c < l

copyC = c + (l - c);

else

copyC = c;

end

% change matrix columns

for ind = 1:copyC

r = yourMAtrix([ind],[col\_1]);

yourMAtrix([ind],[col\_1]) = yourMAtrix([ind],[col\_2]);

yourMAtrix([ind],[col\_2]) = r;

end

% add values to newMat from yourMAtrix

for indl = 1:l

for indc = 1:c

newMat(indl,indc) = yourMAtrix(indl,indc);

end

end

end

**Functia 2**

**Schimbarea pentru ordine crescatoare dupa randul ales**

function [newM] = sortLin(M,lin)

%UNTITLED6 Summary of this function goes here

% Detailed explanation goes here

[l,c] = size(M);

newM = zeros(l,c);

C\_1 = 0;

C\_2 = 0;

min\_idx = 0;

sortat = 1;

for indL = 1:l

for indC = 1:c-1

sortat = 1;

while sortat == 1

sortat = 0;

if M(lin,indC+1) < M(lin,indC)

C\_1 = indC;

C\_2 = indC + 1;

sortat = 1;

M = exe15(M,C\_1,C\_2);

end

end

end

end

% add values to newMat from yourMAtrix

for indl = 1:l

for indc = 1:c

newM(indl,indc) = M(indl,indc);

end

end

end

**SCRIPT**

M = [1 4 2; 4 5 3; 1 4 5; 2 4 6]

lin = 2;

sortLin(M,lin)

**REZULTAT**

>> tema6

M =

1 4 2

4 5 3

1 4 5

2 4 6

ans =

2 1 4

3 4 5

5 1 4

6 2 4

>>

# Exercitiul 16

Să se construiască un fișier de tip funcție prin care să se genereze un vector de dimensiune N+N obținut prin întrețeserea elementelor a doi vectori. Pentru v=[v1,v2,…,vN], w=[w1,w2,…,wN] se obține v\_final=[v1,w1,v2,w2,…,vN,wN]. Vectorii vor fi de aceeași dimensiune, în caz contrar se va afișa un mesaj de eroare

function [finalVect] = exe16(vect1, vect2)

%UNTITLED7 Summary of this function goes here

% Detailed explanation goes here

v1 = length(vect1);

v2 = length(vect2);

fV = v1 + v2;

finalVect = zeros(1,fV);

specialIndex = 0;

if v1 ~= v2

error = 'vectorii trebuie sa fie de aceeasi marime!!!'

elseif v1 == v2

% for vector 1

for index = 1:2:fV

finalVect(index) = vect1(specialIndex + 1);

end

end

% for vector 2

for index = 2:2:fV

finalVect(index) = vect2(specialIndex + 1);

end

end

**SCRIPT**

myVect\_1 = [99,99,99,99,99,99,99,99];

myVect\_2 = [1,1,1,1,1,1,1,1];

exe16(myVect\_2,myVect\_1)

**REZULTAT**

>> tema6

ans =

1 99 1 99 1 99 1 99 1 99 1 99 1 99 1 99

>>

# Exercitiul 18

Şirul lui Fibonacci (fn)n≥0 este definit prin relaţia de recurenţă: fn= fn-1+ fn-2, f0=0, f1=1. Termenii şirului sunt: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, ... Să se construiască o funcţie care afişează numerele lui Fibonacci pentru un n dat. Funcţia va avea ca argument de intrare pe n, iar ca rezultat şirul cu numerele lui Fibonacci. Funcţia va da un mesaj de eroare în cazul în care n este negativ. (error(’mesaj de eroare’);)

function [fibVector] = exe18(n)

%UNTITLED8 Summary of this function goes here

% Detailed explanation goes here

fibVector = zeros(1,n+2);

fibVector(1) = 0;

fibVector(2) = 1;

if n <= 0

error = 'N trebuie sa fie un numar pozitiv!!!'

else

for index = 3:n

fibVector(index) = fibVector(index - 1) + fibVector(index - 2);

end

end

end

**SCRIPT**

n = 9;

exe18(n)

**REZULTAT**

>> tema6

ans =

0 1 1 2 3 5 8 13 21 34 0

>>