Tema Lab. 5

NICOLETA RADU

# Exercitiul 1

Fie matricea ⎥ ⎥ ⎥ ⎦ ⎤ ⎢ ⎢ ⎢ ⎣ ⎡ = − 1 3 4 2 1 3 1 5 7 A . Să se compare rezultatele operaţiilor: A\*A cu A^2 şi inv(A) cu A^(-1) folosind if și să se afișeze un mesaj corespunzător, dacă sunt sau nu echivalente operațiile

A = [1 5 7; 2 1 -3; 1 3 4];

a= A\*A;

b= A^2;

c= inv(A)

d= A^(-1)

if a >= b

rezult\_1 = 'a este mai mare sau egal decat b';

else

rezult\_1 = 'b este mai mare decat a';

end

if c >= d

rezult\_2 = 'c este mai mare sau egal decat d';

else

rezult\_2 = 'd este mai mare decat c';

end

# Exercitiul 2

Calculaţi suma primilor 10 termeni ai vectorului x, având termenul general de forma: xn = (-1)n+1/(2n-1)

a) Calculați suma folosind comenzi Matlab

b) Calculaţi suma folosind instrucţiunea while

c) Calculaţi suma folosind instrucţiunea for

d) Afişaţi rezultatul de la punctul a) cu numere raţionale.

% declarare n si x

n = 1:20;

x = ones(1,20);

s = max(size(x));

for index = 1:1:20

x(index) = ((-1).^ n(index) + 1) / (2 .\* n(index) - 1);

end

x(1:end);

% punctul a

suma\_a = sum(x(1:10));

% punctul b

index = 1;

suma = 0;

while index <= 10

suma = suma + x(index);

index = index +1 ;

end

suma

% punctul c

index = 1;

suma = 0;

for index = 1:10

suma = suma + x(index);

end

suma

% punctul d

rat(suma\_a)

# Exercitiul 3

Să se scrie un fișier script test\_elem.m în Matlab în care se citesc de la tastatură: un vector V şi o valoare a. Să se construiască un vector V1 care conţine toate elementele din V care sunt mai mici decât valoarea a. a) Rezolvaţi problema folosind instrucţiunile de control necesare b) Se cere să se verifice dacă vectorul rezultat, în urma apelului, este vid. (isempty) Indicație: Citirea de la tastatură se face cu input()

V = [];

s = input('introduceti marimea dorita pentru vector: ')

a = input('Introduceti valoare pentru a')

for index = 1:s

V(index) = input('valoare pentru vector: ')

end

index = 0;

V1 = [];

for index = 1:s

if V(index) < a

V1(index) = V(index);

end

end

V1(1:end)

isempty(V1)

# Exercitiul 4

Să se construiască două fișiere funcție pentru calcularea funcțiilor f1 și f2. Să se construiască un fișier script pentru citirea variabilelor x și y de la tastatură și apelarea funcțiilor f1 și f2. Să se afișeze rezultatele funcțiilor f1 și f2 folosind funcția sprintf(). De ce tip de date sunt rezultatele? 2 2 8, 2 1 3 , 2 x dacă x f x dacă x ⎧ + ≤ = ⎨ ⎩ > 2 2 8, 2 3 , 3 2 5 , 4 500, y dacă y y y f y y altfel

%functia f1

function [rezultat1] = f1(x)

% Functia f1

% Calculati daca x <= 2 sau daca x > 2

if x <= 2

rezultat1 = 2 \* x + 8;

elseif x > 2

rezulat1 = 3 \* x^2;

rezultat1 = x;

end

%functia f2

function [rezultat2] = f2(y)

% Functia f2

% Descriere

if y <= 2

rezultat2 = 2 \* y + 8;

elseif y == 3

rezultat2 = 3 \* y^2;

elseif y == 4

rezultat2 = 5 \* y;

else

rezultat2 = 500;

rezultat2 = y;

end

Fisier script

x = input('x egal cu ')

y = input('y egal cu ')

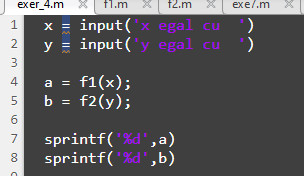
a = f1(x);

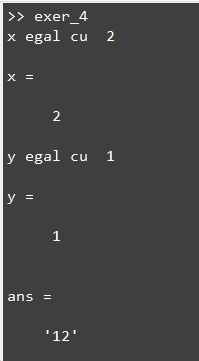
b = f2(y);

sprintf('%d',a);

sprintf('%d',b);

**Rezultatele sunt de tip string (cred)**





# Exercitiul 5

Fie un vector x = [1 4 8 9 11 NaN 14 inf 16 18]. Să se calculeze restul împărțiri la 3 a elementelor vectorului x. Dacă se întâlnește un element care nu este număr finit se cere să se întrerupă calculul. Să se afișeze vectorul cu resturile împărțirii la 3 și un mesaj de eroare dacă s-a oprit calculul în timpul evaluării.

x = [1 4 8 9 11 NaN 14 inf 16 18];

s = max(size(x));

rest = 0;

for index = 1:s

if isfinite(x(index)) == 1

rest = rem(x(index),3)

else

error = 'Numar infinit identificat! Hah!'

break

end

end

# Exercitiul 6

Știind că pentru a cronometra timpul se pot utiliza funcțiile predefinite tic (pornirea cronometrului) și toc (timpul la oprirea cronometrului) se cere să se cronometreze timpul necesar pentru execuția următoarelor cerințe:

a) Să se genereze matrice pătratice de numere aleatoare cu distribuție uniformă în domeniul (0,val), unde val=1,2,…,1000. Matricea are dimensiunea 500x500. Software Computaţional – Laborator 5 Conf. dr. Laura Stoica 9

b) Să se genereze vectori cu 100 elemente echidistante în domeniul [val,val+5], unde val=1,2,…,1000. Știind că timpul este afișat în secunde, determinați timpul pentru instrucțiunile for respectiv while fără afișarea valorilor din matricele create. Indicație: tic %pornire ceas for i=1:2:1000 end toc %oprire ceas Rezultatul la execuția comenzii toc este: Elapsed time is 0.000165061 seconds.

% punctul a

arr = zeros(500);

s = size(arr);

val = 1;

tic

for indc = 1:s

for indl = 1:s

if val < 1000

arr([indc],[indl]) = val \* rand(1);

else

val = 0;

end

val = val + 1;

end

end

toc

Elapsed time is 0.022761 seconds.

>>

**!!** **Nu stiu daca am inteles bine cerinta problemei, dar alta abordare asupra exercitiului ar fi urmatoarea:**

% punctul a

**Interval (0,val)**

tic

for val = 1:1:999

val \* rand(500);

end

toc

>> exer\_4

Elapsed time is 1.716922 seconds.

# Exercitiul 7

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

• x=1, dacă există în matricea A cel puțin un element mai mare decât modulul diferenţei dintre suma elementelor de pe diagonala principală şi suma elementelor de pe diagonala secundară şi

• x=0 în caz contrar.

function [x] = exe7(arr)

% Exercitiul 7

% Input - matrice, output - valoare

x = 0;

[l,c] = size(arr);

diagonalaA = sum(diag(arr));

diagonalaB = sum(diag(fliplr(arr)));

rezultat = abs(diagonalaA - diagonalaB);

if l == c

for indl = 1:1:l

for indc = 1:1:c

if arr([indl],[indc]) > rezultat

x = 1;

break

end

end

end

else

error = 'Matricea nu este patratica'

x = arr;

end

A1=[1 2 3 ; -1 1 2 ; 1 2 1 ];

rezultfunc1 = exe7(A1)

A2=[1 2 3; 4 -5 6; 7 -8 -1];

rezultfunc2 = exe7(A2)

>> exer\_4

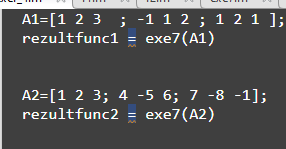
rezultfunc1 =

1

rezultfunc2 =

0

>>



# Exercitiul 8

Să se scrie un program în Matlab unde se citesc de la tastatură un vector V şi o valoare “a”. Programul va afișa un vector Vpar care conţine toate elementele pare din V care sunt mai mari decât valoarea “a”. Să se rezolve problema prin două metode (1. comenzi Matlab, 2. instrucţiuni de control).

s = input('Dimensiune vector: ')

a = input('Valoare variabila a: ')

V = [];

for index = 1:s

V(index) = input('V[] = ')

end

% comenzi Matlab

Vpar = [];

Vpar = V(rem(V,2) == 0 & V > a);

% instructiuni de control

s = size(V);

for index = 1:s

if rem(V,2) == 0

if V(index) > a

Vpar(index) = V(index);

end

end

end

Vpar

# Exercitiul 9

Să se scrie un program în Matlab unde se citesc de la tastatură două matrice A şi B, de aceeaşi dimensiune. Programul va afișa valoarea lui v definită astfel:

• v=1 - dacă toate elementele din A sunt mai mici decât toate elementele din B

• v=2 - dacă toate elementele din A sunt mai mari decât toate elementele din B

• v=3 - în caz contrar

A\_l = input('Dimensiune matrice A, linii: ')

sA\_c = input('Dimensiune matrice A, coloane: ')

sB\_l = input('Dimensiune matrice B, linii: ')

sB\_c = input('Dimensiune matrice B, coloane: ')

A = [];

B = [];

cot = 100;

while cot > 1 && sA\_l ~= sB\_l && sA\_c ~= sB\_c

warning = 'Matricele trebuie sa fie de aceleasi dimensiuni!'

sA\_l = input('Dimensiune matrice A, linii: ')

sA\_c = input('Dimensiune matrice A, coloane: ')

sB\_l = input('Dimensiune matrice B, linii: ')

sB\_c = input('Dimensiune matrice B, coloane: ')

cot = cot - 1;

end

% Valori A

for indexL = 1:sA\_l

for indexC = 1:sA\_c

A([indexL], [indexC]) = input('A[] = ')

end

end

% Valori B

for indexL = 1:sB\_l

for indexC = 1:sB\_c

B([indexL], [indexC]) = input('B[] = ')

end

end

% verificare v=1

message = '\*\*\*\*\*\*\*\*\*\*\*v1\*\*\*\*\*\*\*\*\*'

for indexL = 1:sA\_l

for indexC = 1:sA\_c

if A([indexL], [indexC]) > B([indexL], [indexC])

v = 3;

break

else

v = 1;

end

end

end

v

% verificare v=2

message = '\*\*\*\*\*\*\*\*\*\*\*v2\*\*\*\*\*\*\*\*\*'

for indexL = 1:sA\_l

for indexC = 1:sA\_c

if A([indexL], [indexC]) < B([indexL], [indexC])

v = 3;

break

else

v = 2;

end

end

end

v

# Exercitiul 10

Fie dată o matrice A = [1, 2, 3; 2, 1, 5; 4, 6, 4; 2, 3, 2]. Creați o matrice nouă B unde pe fiecare linie se va păstra valoarea maximă, iar restul de elemente vor fi modificate în 0.

A = [1, 2, 3; 2, 1, 5; 4, 6, 4; 2, 3, 2]

[l,c] = size(A);

B = zeros(l,c);

maximum = 0;

copyL = 0;

copyC = 0;

for indexL = 1:l

for indexC = 1:c

if A([indexL],[indexC]) > maximum

maximum = A([indexL],[indexC]);

copyL = indexL;

copyC = indexC;

end

end

B([copyL],[copyC]) = maximum; % schimba valoarea cu maximuml

maximum = 0;

end

B

# Exercitiul 11

Fie dată o matrice A = [1, 2, 3; 2, 1, 5; 4, 6, 4; 2, 3, 2]. Creați o matrice nouă C unde pe fiecare coloană valoarea minimă se va modifica în 0, iar la restul de elemente se va schimba semnul.

A = [1, 2, 3; 2, 1, 5; 4, 6, 4; 2, 3, 2]

[l,c] = size(A);

C = zeros(l,c);

minimum = 99999;

copyL = 0;

copyC = 0;

for indexL = 1:l

for indexC = 1:c

if A([indexL],[indexC]) < minimum

minimum = A([indexL],[indexC]);

copyL = indexL;

copyC = indexC;

end

C([indexL],[indexC]) = A([indexL],[indexC]) .\* -1;

end

C([copyL],[copyC]) = 0; % schimba valoarea minima in 0

minimum = 99999;

end

C

# Exercitiul 12

Să se creeze un fişier script cautasir.m în care se cunosc șirurile sir=’abaanamarianaab’ şi subsir=’ana’. Să se caute subsir în sir și să se înlocuiască apariţiile lui subsir cu un şir de blancuri şi să se afişeze noul şir de caractere modificat.

sir = 'abaanamarianaab';

subsir= 'ana';

newCh = strrep(sir,subsir,' ')

newCh =

'aba mari ab'

>>

# Exercitiul 13

Fie șirul de caractere 'Doresc-sa-impart-acest-sir-de-caractere'. Se cere scrierea șirului de caractere în 7 subșiruri separate (fiecare cuvânt va fi scris pe o linie nouă – str2mat).

Se cunoaște că fiecare subșir este separat prin același caracter ’-’ (minus). Rezolvați problema utilizând instrucțiuni de control for sau while și funcția strtok.

Cronometrați timpul necesar pentru execuția programului și afișarea rezultatului. (tic, toc) Indicație: • strtok(’sir de caractere separat prin spatii’,’ ’) o al doilea argument reprezintă caracterul de separare a cuvintelor. Funcția returnează primul subșir găsit până la token-ul căutat (caracterul de separare a cuvintelor) • [subsir1,subsirul\_ramas]=strtok(’sir de caractere separat prin spatii’,’ ’) Vezi help strtok

**Incercare nereusita (printre multe altele):**

sir = 'Doresc-sa-impart-acest-sir-de-caractere';

s = length(sir)

[token,remain] = strtok(sir, '-');

A = str2mat(token) % am pus 'Doresc' prima oara, apoi restul in while loop

tic

while remain ~= 0

[token,remain] = strtok(remain, '-');

A = str2mat(A,token);

end

toc

A(1:end,:)

# Exercitiul 14

Să se scrie un fișier script în care se citește de la tastatură un vector V. Rezultatul în urma prelucrărilor vectorului V va fi o matrice A cu două coloane, care să conţină elementele din V până la apariţia primului element mai mic sau egal cu 0. Elementele vor fi aşezate în matricea A linie după linie. Dacă numărul de elemente având proprietatea cerută este impar se va introduce la final un element egal cu 0. Dacă matricea A este vidă să se afișeze un mesaj corespunzător.

**Solutie aproximativ corecta**

dimensiune = input('Dimensiune vector V: ')

V = [];

for index = 1:dimensiune

V(index) = input('V[] = ')

end

A = zeros(round(dimensiune/2),2);

[l,c] = size(A)

index2 = 1;

for indexL = 1:l

for indexC = 1:c

if V(index2) > 0

A([indexL],[indexC]) = V(index2);

index2 = index2 + 1;

else

A([indexL],[indexC]) = 0;

break

end

end

end

if isempty(A) == 1

message = 'Matricea contine numai valori nule'

end

A(1:end,:)

# Exercitiul 15

Se citeşte de la tastatură un număr format din mai multe cifre. Se cere să se verifice dacă numărul este palindrom. (Un număr este palindrom dacă citit de la stânga la dreapta sau de la dreapta la stânga rămâne neschimbat. Ex: 123321, 14541, 11211211). Rezolvați problema prin două moduri: (1) convertind numărul în șir; (2) ca un vector de cifre (Indicație: ultima cifra se obține prin rem(x,10), determinăm restul, iar pentru cât avem fix(x/10)).

numar1 = input('Introduceti un numar: ')

copy1 = numar1;

inversNumar = 0;

adev = 1;

while copy1 > 0

inversNumar = inversNumar \* 10 + rem(copy1,10);

copy1 = fix(copy1/10);

end

inversNumar

while inversNumar > 0 && numar1 > 0 && adev == 1

if rem(inversNumar,10) == rem(numar1,10)

adev = 1;

else

adev = 0;

end

inversNumar = fix(inversNumar/10);

numar1 = fix(numar1/10);

end

adev

# Exercitiul 16

Se citeşte de la tastatură o matrice pătratică A. Să se scrie fiecare element cu negativul lui cu excepţia elementelor de pe diagonala principală. Se va testa dacă matricea este pătratică. Dacă matricea citită nu este pătratică atunci se va adăuga 0 pe linii sau pe coloane pentru a forma matricea pătratică.

Al = input('Dimensiune matrice A, linii: ')

Ac = input('Dimensiune matrice A, coloane: ')

A = zeros(Al,Ac);

missing = 0;

% Valori A

for indexL = 1:Al

for indexC = 1:Ac

A([indexL], [indexC]) = input('A[] = ')

end

end

A(1:end,:)

if Al < Ac % daca numarul liniilor este mai mic

missing = Ac - Al;

backup = zeros(missing,Ac);

A = [A ; backup];

elseif Ac < Al % daca numarul coloanelor este mai mic

missing = Al - Ac;

backup = zeros(Al,missing);

A = [A , backup]

end

A(1:end,:)

[Al,Ac] = size(A);

for indexL = 1:Al

for indexC = 1:Ac

if indexC ~= indexL % schimbare valori mai putin pe diagonala principala

A([indexL],[indexC]) = A([indexL],[indexC]) \* -1;

end

end

end

A(1:end,:)

**ans =**

**354 567 879 809**

**9 687 756 456**

**432 645 756 798**

**534 324 564 756**

**ans =**

**354 -567 -879 -809**

**9 687 -756 -456**

**-432 -645 756 -798**

**-534 -324 -564 756**

# Exercitiul 17

Se cere să se genereze o matrice A de dimensiune MxN, unde M,N sunt două numere naturale aleatoare în domeniul [2,5], iar elementele matricei sunt numere naturale în domeniul [0,30]. Să se construiască un vector linie V, ale cărui elemente V(i) să fie date de cel mai mare divizor comun al elementelor de pe coloana i a matricei A. (Indicație: lin=fix(2+3\*rand(1))).

N = randi([2,5],1);

M = randi([2,5],1);

A = randi([0,30],M,N)

**Incercare esuata... printre multe altele:**

V = [];

indexC = 1;

for indexL = 1:M

divizorComun = gcd(A(indexL,indexC),A(indexL+1,indexC))

for indexC = 1:N

divizorComun = gcd(divizorComun,A(indexC,indexL));

end

V(indexC) = divizorComun;

end

V(1:end)

# Exercitiul 18

(Opțional) Să se scrie un program pentru crearea a n vectori linie de forma: v1=[1], v2=[1,2], v3=[1,2,3], … vk=[1,2,…,k],… vn=[1,2,…,n]. Se cere construirea unei matrice de dimensiune nxn unde pe prima linie se găsesc elementele lui v1, pe a doua linie elementele lui v2, etc.

Pentru a construi matricea este nevoie ca fiecare loc liber să fie completat cu elemente de 0. Adică prima linie va avea n-1 zerouri, a doua linie n-2 zerouri, etc. Variabila n este un număr natural din domeniul [4,9]. Indicație: Pentru n=4, matricea va fi : 1 0 0 0 1 2 0 0 1 2 3 0 1 2 3 4 Pentru construirea vectorului v se folosește funcția eval cu forma: eval(['v',num2str(indice),'=1',':',num2str(indice)]) Astfel s-au creat vectorii v1, v2, v3, …