

Seminarul 1

PROGRAMARE ÎN SCILAB

SCILAB-ul poate lucra în următoarele moduri:

- În modul linie de comanda; fiecare linie este prelucrată imediat și rezultatele sunt afișate;
- Cu programe conținute în fișiere. Un fișier constă dintr-o succesiune de instrucțiuni SCILAB, cu posibilitatea apelării altor fișiere precum și a apelării recursive. Un program SCILAB poate fi scris sub forma fișierelor:
 - **Script:** conține o secvență de comenzi SCILAB; se execută în fereastra de comandă prin apelarea numelui fișierului: `exec('nume_fisier')` sau direct, în editorul Scilab-ului folosind din meniul 'Execute' opțiunea 'Load into Scilab'. **Funcție:** prima linie conține cuvântul *function*, iar ultima linie conține cuvântul *endfunction*; poate lucra cu argumente. Se pot defini mai multe funcții în același fișier.

Sintaxa instrucțiunilor SCILAB

Instrucțiune	Sintaxă
if	if <i>expresie logică</i> grup instrucțiuni end
else	if <i>expresie logică</i> grup instrucțiuni A else grup instrucțiuni B end
elseif	if <i>expresie logică_1</i> grup instrucțiuni A elseif <i>expresie logică_2</i> grup instrucțiuni B end
case	select expr_0 case expr_1 Instr_1 ... case expr_n Instr_n else Instr end
for	for index = expresie \\ expresie = inițial: pas: final grup_instrucțiuni end
while	while expresie grup_instrucțiuni end

Exercitii

1. Sa se genereze o matrice A, cu n linii si n+1 coloane, ale carei elemente sunt:

$$A = \begin{cases} 2, & \text{daca } i = j \\ -1, & \text{daca } |i - j| = 1 \\ 0, & \text{in rest} \end{cases}$$

Solutie:

```
n=10;
for i=1:n
for j=1:n+1
if i==j
A(i,j)=2;
elseif abs(i-j)==1;
A(i,j)=-1;
else
A(i,j)=0;
end
end
end
```

A

2. Sa se scrie un program utilizand **while** care calculeaza suma elementelor vectorului $x=(5 \ 2 \ -9 \ 10 \ -1 \ 9 \ -1)$ pana cand intalneste un nr mai mare ca 8.
3. Să se calculeze limitele următoarelor șiruri definite recurent:

a) $\lim_{n \rightarrow \infty} c_n$, unde $c_1 = 10$ și $c_{n+1} = \frac{2 + c_n^2}{2c_n}, \forall n \geq 1$;

Solutie:

Se scrie programul în fișierul, de exemplu *SirRecurent.sce*:

```
x=10;
c=(2+x^2)/(2*x);
while abs(c-x)>0.00001
    x=c;
    c=(2+x^2)/(2*x);
end
c
```

Se lansează în execuție în fereastra de comandă scriind:

```
-->exec("SirRecurent.sce")
```

și se obține rezultatul:

```
c = 1.4142136
```

b) $\lim_{n \rightarrow \infty} x_n$, unde $x_0 = 1$ și $x_{n+1} = \frac{x_n}{\sqrt{x_n^2 + 1}}, \forall n \geq 0$

4. Sa se calculeze si sa se afiseze $A = \sum_{k=1}^{10} (2k-1)^3 + \prod_{k=1}^{10} 3k^2$

5. Să se calculeze suma elementelor unui vector care sunt cuprinse între valorile **a** și **b** cu $a < b$.

Solutie:

```
v=[2 5 7 -2 5 8 0 12 32];
s=0;
a=-1;b=7;
for i=1:length(v)
    if (v(i)>a)&(v(i)<b)
        s=s+v(i);
    end
end
s
```

Se accesează meniul **Execute/Load into Scilab**, soluția va apărea în fereastra de comandă:

```
--> s = 12.
```

6. Să se calculeze produsul elementelor diferite de zero ale unui vector.
7. Se dă șirul de numere x_1, x_2, \dots, x_n . Să se determine numărul de elemente pozitive și să se calculeze produsul lor.
8. Să se calculeze media aritmetică a elementelor unui vector care sunt mai mari decât o valoare dată.
9. Se dă șirul de numere x_1, x_2, \dots, x_n . Să se calculeze:
 - media aritmetică a numerelor pozitive;
 - suma pătratelor numerelor negative
10. Să se ordoneze crescător elementele unui vector.
11. Să se calculeze produsul scalar al vectorilor $x = (x_1, x_2, \dots, x_n)$ și $y = (y_1, y_2, \dots, y_n)$.

12. Se dau două siruri de câte n numere: a_1, a_2, \dots, a_n și b_1, b_2, \dots, b_n . Să se formeze șirul c_1, c_2, \dots, c_n în care termenii se obțin după următorul procedeu: pentru orice $i, i = \overline{1, n}$,

$$c_i = \begin{cases} \frac{a_i + b_i}{2}, & \text{daca } a_i \cdot b_i < 0 \\ \max\{a_i, b_i\}, & \text{daca } a_i \cdot b_i \geq 0 \end{cases}$$

FIȘIERE DE TIP FUNCȚIE

Fișierele funcție sunt utilizate pentru crearea unor noi funcții SCILAB. Forma generală a unui fișier funcție este:

```
function [parametri_iesire] = nume_funcție (parametri_intrare)
.....instrucțiuni de definire a funcției....
endfunction
```

unde:

function: cuvânt cheie care declară fișierul ca fișier funcție (obligatoriu);

nume_funcție: numele funcției, adică numele sub care se salvează fișierul, fără extensie; nu poate fi identic cu cel al unui fișier existent;

parametri_iesire: parametrii de ieșire trebuie separați cu virgulă și cuprinși între paranteze drepte. Dacă funcția nu are parametri de ieșire parantezele drepte și semnul egal (=) nu mai au sens;

parametri_intrare: parametrii de intrare trebuie separați cu virgulă și cuprinși între paranteze rotunde. Dacă funcția nu are parametri de intrare parantezele rotunde nu mai au sens.

endfunction: instrucțiunea de terminare a unei funcții.

Observații:

1. Un fișier de tip funcție poate să contină mai multe funcții. Pentru utilizarea unei funcții trebuie folosită, mai întâi, comanda:

```
getf("NumeFuncție.sci")
```

2. Spre deosebire de fișierele *script* care au extensia 'sce', fișierele de tip funcție au extensia 'sci'.
3. O funcție se poate defini direct prin intermediul comenzii *deff*

```
deff('[y1, y2, ..., ]=nume_funcție(x1, x2, ...)',text)
```

Exemplu: `deff('[y]=f(x)', 'y=sin(x).*cos(x)')`

EXERCITII

1. Sa se calculeze si sa se afiseze suma patratelor elementelor mai mari decat 1 ale unei matrice patratice.

Solutie:

```
function [s]=sumaP(a)
[m,n]=size(a)
s=0;
for i=1:m
    for j=1:n
        if a(i,j)>=1
            s=s+a(i,j)^2
```

```

end
end
end
endfunction

```

2. Sa se calculeze media aritmetica a elementelor de pe diagonala principala a unei matrice patratice de ordinul n , cu $n < 10$
3. Sa se determine valorile maxime si minime ale unui vector si ale unei matrice, apoi sa se determine si indicele (pozitia acestora) (se vor folosi functiile Scilab-ului).

4. Sa se determine valorile si vectorii proprii ale matricei
$$\begin{pmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix}.$$

Solutie:

In determinarea solutiilor nebanale ale ecuatiei $Ax = \lambda x$ unde:

A – este o matrice patratice de ordinul n ,

x – este vector coloana de ordinul n , iar

λ - este un scalar,

valorile x , respectiv λ , care satisfac ecuatia de mai sus se numesc vectori proprii respectiv valori proprii.

```
-->a=[-2 1 0;1 -2 1;0 1 -2];
```

```
-->[v,d]=spec(a)
```

```

d =! - 3.4142136  0.      0.      !
      !  0.      - 2.      0.      !
      !  0.      0. - 0.5857864 !
v =!  0.5      - 0.7071068  - 0.5      !
      ! - 0.7071068  2.082D-17 - 0.7071068 !
      !  0.5      0.7071068  - 0.5      !

```

5. Să se calculeze suma elementelor diferite de o valoare dată, v , ale unei matrice.

Solutie:

```

function [s]=sdv(a,v)
[m,n]=size(a)
s=0;
for i=1:m
    for j=1:n
        if a(i,j)~=v
            s=s+a(i,j)
        end
    end
end
end
endfunction

```

6. Sa se determine elementul maxim al unui matrice de numere reale (se va defini o functie in editorul scilab-ului).

7. Să se calculeze produsul elementelor diferite de zero de pe diagonala principală (secundară) ale unei matrice.
8. Să se calculeze suma elementelor unei matrice situate deasupra (dedesubtul) diagonalei principale (secundare).
9. Sa se realizeze un program care calculeaza produsul a doua matrice.
10. Se dau numerele $a_{ij}, i = 1, 2, \dots, 100; j = 1, 2, \dots, 50$ si $b_j, j = 1, 2, \dots, 50$. Sa se realizeze un program care calculeaza:
$$c_i = \sum_{j=1}^{50} a_{ij} b_j, i = 1, 2, \dots, 100.$$