

Laborator #8

18 nov. 2009

1. Recapitulare: 1.1. Folosind for și if, calculați valorile funcției $f(x)=x^2+x+2$ pt x din intervalul $[-10, 1]$ și $f(x)=3x+1$ pt x din intervalul $(1, 15]$.

1.2. Scrieți un ciclul for care pentru index= expresie conține pe poziția expresiei un vector de valori și un al doilea caz când expresie este o matrice. Urmăriți parcurgerea elementelor matricei.

2. Instrucțiunea de ciclare **while** cu condiționare anterioară

while <i>expresie logica</i> bloc de instrucțiuni end	Se execută un grup de instrucțiuni de un număr nedefinit de ori, sub controlul unei condiții logice
i=-2; while i<7 i=i+1; end i	%suma elementelor x(2)... x(5) din sirul x x=1:10; suma=0; i=2; while i<=5 suma=suma+x(i); i=i+1; end suma
%Suma elementelor negative x=[1 -2 3 -4 -5 6]; suma=0; %initializam variabila suma i=1; %initializam „contorul” i n_e= length(x) while i <= n_e if x(i)<0 suma=suma + x(i); end i=i+1; end suma	%instrucțiune while in corpul while m=input('linii m= '); n=input('coloane n= '); i=1; while i<=m %se repeta de m ori j=1 while j<=n %se repeta de m x n ori A(i,j) = 1/(i + j - 1); j=j+1; end i=i+1; end i j plot(A); % este mai dificila gestionarea indicilor %decât în cazul folosirii ciclului for

3. Instrucțiunea **break** realizează întreruperea ciclării **for** sau **while** și salt la prima instrucțiune după instrucțiunea **end** a ciclării.

4. Instrucțiunea **continue** trece controlul sau realizează salt la următoarea iterație în cadrul ciclărilor for sau while.

5. Instrucțiunea **return** realizează ieșirea forțată din funcție (funcția apelată fA) și întoarcerea la funcția apelantă fAn (care a apelat procedura).

function out=fun_demo1(in) i=1; while i < in x(i)=rand(1); if x(i)>0.95 'break' break; end i=i+1; end out=x; i display('exemplu cu break') end %incheiere functie cu end optional	function out=fun_demo2(in) i=1; while i < in x(i)=rand(1); if x(i)>0.95 out=x; %salvam x in out i 'return' return; end i=i+1; end out=x; display('exemplu cu return') i
--	---

6. Definirea unei funcții în Matlab

6.1. Sintaxa definirii unei funcții:

function [out1, out2, ...outm] = nume_fun(in1, in2, ...inn) grup de instructiuni (corpul functiei) out1=... outm=... end % end optional
APEL functie: [r1, r2,...,rm]=nume_fun(a1, a2,...,an)

6.2. EXEMPLE

1) Un parametru de intrare și unul de ieșire

function y = average(x) % AVERAGE media elementelor vectorului y = sum(x)/length(x); end	Funcția average salvată în fișierul average.m
---	---

Exersați APELURI valide ale funcției:

v=[1 2 3 -3 5]; ave=average(v)	v=[1 2 3 -3 5]; average(v) ans = 1.6000	average([1 2 3 -3 5]) ans = 1.6000
-----------------------------------	---	--

2) Doi parametri de intrare și 1 parametru de ieșire

function z=xplusy(x,y); z=x+y; end
--

APELURI valide ale funcției:

a=1 b=2 scal1=xplusy(a,b) scal2=xplusy(1,2)	a=[1 2 3; 3 4 5] b=[7 8 9; 2 0 4] matr1=xplusy(a,b) matr1=xplusy([1 2 3; 3 4 5],b)	a=7 b=[7 8 9; 2 0 4] matr2=xplusy(a,b) matr2=xplusy(7, [7 8 9; 2 0 4])
--	---	---

3) Doi parametri intrare, fără parametri de ieșire

function comp(a,b) if a>b 'a>b' elseif a<b 'a<b' else %a==b 'egalitate sau alta situatie' end	a,b parametri intrare; nu exista parametri de ieșire Salvam definitia în fișierul comp.m APELURI: >>comp(4,6)
---	--

4) Un parametru de intrare, 2 parametri de ieșire

function [s,sp]=sumap(x) %returneaza suma si suma elem >0 s=0;sp=0; for i=1:length(x) s=s+x(i); if x(i)>0 sp=sp+x(i); % insumare % x(i)>0 end end end	function [mean2,stdev] = stat2(x) n = length(x); mean2 = sum(x)/n; stdev = sqrt(sum((x-mean2).^2) /n); end
Apel: [suma, sp]=sumap([1 5 -3])	

5) Scrieți o funcție care conține o **ciclare** și instrucțiunea **continue** plasată în corpul ciclului.

I. Lupea