

Laborator #9

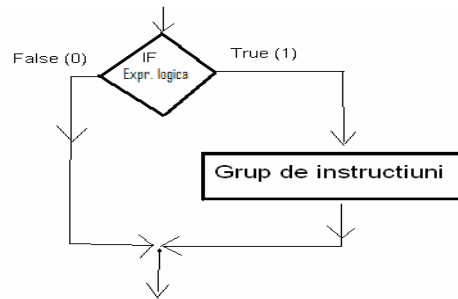
1. Continuare funcții:

Pentru evaluarea unei formule (funcții) se pot folosi **funcții inline** pentru a evita definirea unei funcții.

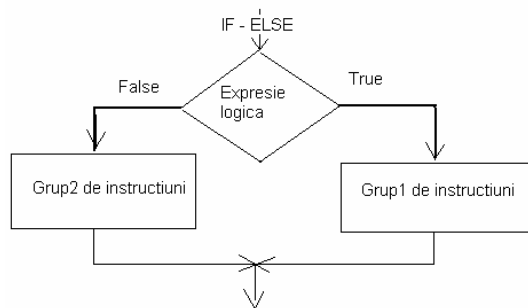
function y=f_expresie(x) y=x+sin(x)+x.^2;	fun_ex=inline(x+sin(x)+x.^2); %arg. sin radiani
Sunt permise apelurile:	
Re1=f_expresie(2.2)	Re2=fun_ex(2.2)
Re1=f_expresie([2.2 3 5])	Re2=fun_ex([2.2 3 5])

2. Scheme logice asociate instrucțiunilor

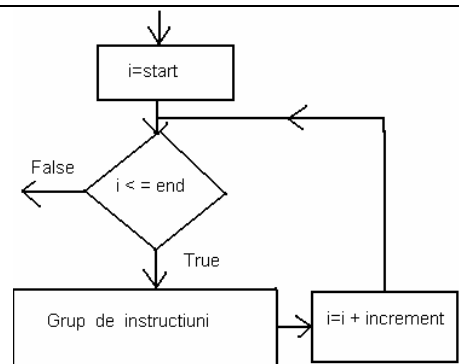
```
if expresie_logica
    grup de instructiuni
end
```



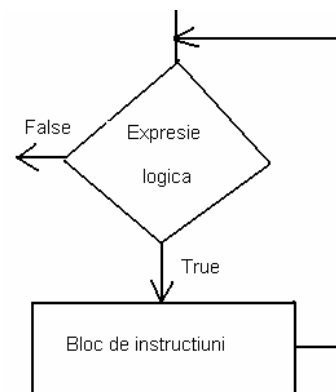
```
if expresie_logica
    grup1 de instructiuni
else
    grup2 de instructiuni
end
```



```
for index = start:increment:end
    grup de instructiuni
end
```



```
while expresie logica
    bloc de instructiuni
end
```



switch expresie

% scalar sau sir de caractere/string

case valoare1

grup1 instructiuni

case valoare2

grup 2 instructiuni

.

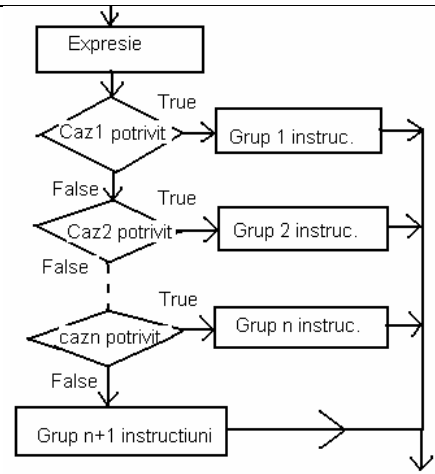
case valoare n

grupn instructiuni

otherwise

grup n+1 instructiuni

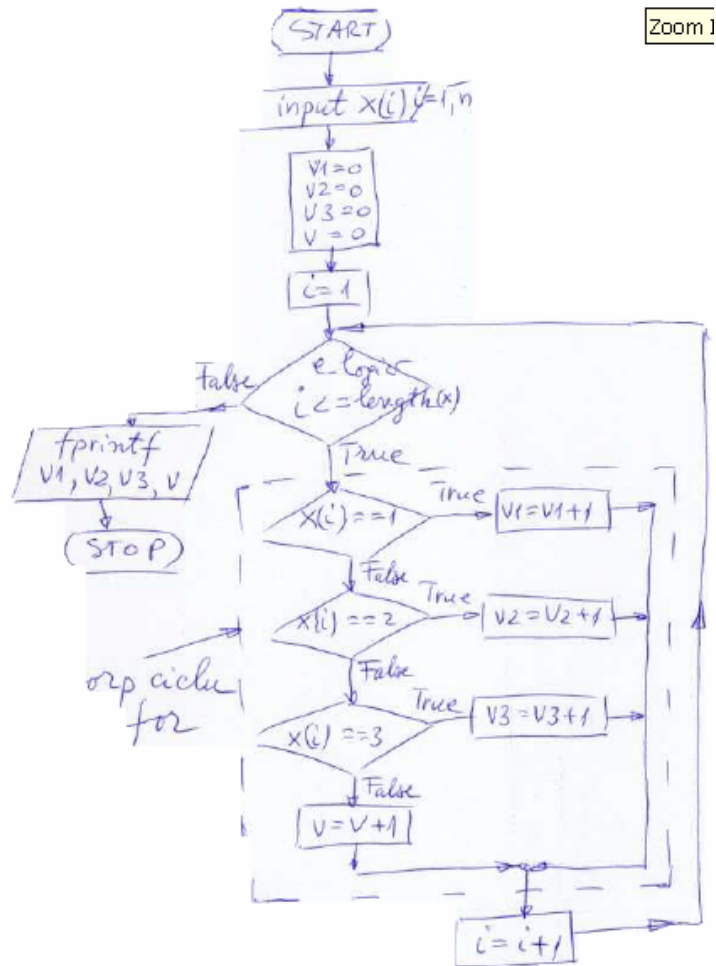
end



3. Să se traseze schema logică asociată programului de mai jos.

```

%contorizeaza numarul de valori
% 1 2 3 si restul valorilor
x=input('sir de valor x= ');
v1=0; v2=0; v3=0; v=0;
for i=1:length(x)
    if x(i)==1
        v1=v1+1;
    elseif x(i)==2
        v2=v2+1;
    elseif x(i)==3
        v3=v3+1;
    else
        v=v+1;
    end
end
fprintf('v1=%d v2=%d v3=%d
v=%d\n',v1,v2,v3,v);
  
```



4. Trasați scheme logice asociate unor programe deja parcurse la laborator și pentru noi programe.