MATLAB

Logičke operacije i operacije poređenja

Kontrola toka programa

Logički i operatori poređenja

Operator	Operacija
&	Logičko I
|	Logičko ILI
~	Logička negacija

Operator	Operacija
>	Veće od
>=	Veće od ili jednako
<	Manje od
<=	Manje od ili jednako
==	Provjera jednakosti
~=	Provjera nejednakosti

Prvenstvo operatora

Operator
()
.' .^ ' ^
unarni + unarni - ~
.* ./ * /
+ -
:
< <= > >= == ~=
&
|

Pri vrhu tabele su operatori sa većim prioritetom, a pri dnu oni sa manjim prioritetom.

Kontrola toka programa – Pregled naredbi

- Naredbe uslovnog skoka:
  - `if`
  - `if-else`
  - `if-elseif-else`
- Programske petlje:
  - `for` petlja
  - `while` petlja

Naredba if

- Sintaksa naredbe:

```
if (uslov)
   naredbe
end
```

Ključna riječ `end` označava kraj `if` naredbe

- Ukoliko je logički uslov ispunjen izvršavaju se naredbe, u suprotnom se ide na prvu naredbu poslije `end`.

- Primjer:

```
n = input('Unesi broj ');
if (n > 10)
  disp('n je vece od 10');
end
```

`input` funkcija za unos podatka

`disp` funkcija za štampanje teksta

Naredba if-else

- Sintaksa naredbe:

```
if (uslov)
   naredbe1
else
   naredbe2
end
```

- Ukoliko je logički uslov ispunjen izvršavaju se naredbe1, u suprotnom se izvršavaju naredbe2.

- Primjer:

```
n = input('Unesi broj ');
if (n > 10)
  disp('n je vece od 10');
else
  disp('n je manje ili jednako 10');
end
```

Naredba if-elseif-else

- Sintaksa naredbe:

```
if (uslov1)
    naredbe1
elseif (uslov2)
    naredbe2
...
else
    naredbeK
end
```

Uslovi se provjeravaju redom i izvršavaju se prve naredbe čiji je uslov ispunjen. Nakon toga se nastavlja sa naredbama nakon ključne riječi end.

- Primjer:

```
n = input('Unesi broj ');
if (n > 10)
   disp('Broj je veci od 10');
elseif (n < 3)
   disp('Broj je manji od 3');
else
   disp('Broj je izmedju 3 i 10');
end
```

Programske petlje – for petlja

- U MATLAB-u mogu da se koriste **`for`** i **`while`** petlje.
- `for` petlja (ili brojačka petlja) se koristi kada se zna koliko puta treba da se izvrši određeni dio koda (tijelo petlje).
- Sintaksa `for` petlje:

```
for i = a:b:c
    naredbe
end
```

gdje je:

`i` – brojačka promjenljiva

`a` – početna vrijednost brojača `i`

`b` – korak promjene brojača `i`

`c` – krajnja vrijednost brojača `i`

```
for i = 1 : 2 : 10
   disp(i)
end
     1
     3
     5
     7
     9
```

Primjer sa for petljom

- Napisati m-fajl **Niz** kojim se unosi niz **X** i broj **N** i koji određuje i štampa koliko se puta broj N pojavljuje u nizu X.

```
X = input('Unijeti niz X ');
N = input('Unijeti broj N ');
Br = 0;
for i = 1 : length(X)
   if X(i)==N
      Br = Br + 1;
   end
end
if Br==0
   disp('Nema ga!');
else
   disp('Broj pojava je');
   disp(Br);
end
```

Jedno izvršenje

```
>> Niz
Uneti niz X [1,2,3,4,3,2,4,5,6,5]
Uneti broj N 5
Broj pojava je
     2
```

Drugi primjer sa for petljom

- Napisati funkciju **PozNegNul** koji za ulazni argument ina niz **X** i koji vraća broj pozitivnih elemenata, broj negativnih elemenata i broj nula u nizu X.

```
function [brPoz,brNeg,brNul] = PozNegNul(X)
brPoz = 0; brNeg = 0; brNul = 0;
for i = 1 : length(X)
    if X(i) > 0
        brPoz = brPoz + 1;
    elseif X(i) < 0
        brNeg = brNeg + 1;
    else
        brNul = brNul + 1;
    end
end
```

Jedno izvršenje

```
>> X = [1,0,-3,4,3,-2,4,0,6,5];
>> [brPoz,brNeg,brNul] = PozNegNul(X)

brPoz =

     6


brNeg =

     2


brNul =

     2
```

Treći primjer sa for petljom

- Napisati funkciju **obrni** koji za ulazni parametar ima niz brojeva **X** i koja vraća niz koji ima obrnut redosled elemenata u odnosu na niz X.

```
function Y = obrni(X)
Y = [];
for i = 1 : length(X)
    Y = [X(i), Y];
end
```

Jedno izvršenje

```
>> obrni([6,5,4,3,0,1])

ans =

     1     0     3     4     5     6
```