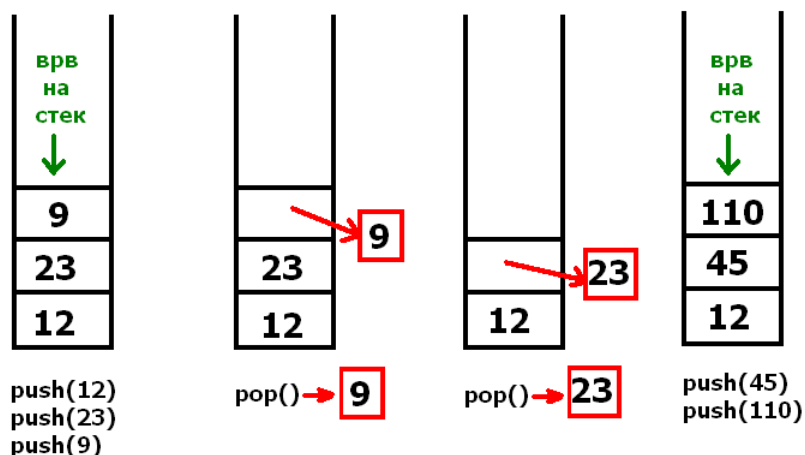


Податочна структура: Стек

Стект е една од основните податочни структури што се користи при програмирањето. Оваа структура се реализира со помош на низа што исто така претставува посебна и добро позната податочна структура. Стект може да се сфати како низа со поставени правила на работа (употреба) на таа низа:

- Поставување елемент на врвот на стект : *push(x :element)*
- Читање и одстранување на елементот кој се наоѓа на врвот од стект : *pop()*

Стект има свој капацитет. При ставање на нов елемент на врвот од стект, (се проверува) се внимава да не се надмине капацитетот на стект. Но исто така не можеме да извадиме елемент од стект доколку нема ниту еден елемент којшто е поставен во стект. Стект припаѓа на **LIFO**(*Last In First Out*) структурите.



Пример програма за употреба на стект:

```
program primer_za_stek;
var
    stek:array[1..100]of integer;
    kapacitet:integer;
    tekoven_broj_elementi:integer;

procedure inicijaliziraj_stek;
begin
    kapacitet:=100;
    tekoven_broj_elementi:=0;
end;

procedure push(x:integer);
begin
    {proveruvame dali ima se uste mesto vo stekot}
    if tekoven_broj_elementi<kapacitet
    then
        begin
            inc(tekoven_broj_elementi);
            stek[tekoven_broj_elementi]:=x;
        end
    else
        writeln('Nema poveke mesto vo stekot !');
    end;
end;

function pop():integer;
```

```

begin
  {proveruvame dali ima barem eden broj postaven vo stekot}
  if tekoven_broj_elementi>0
    then
      begin
        {go vrakame brojot od vrvot na stekot kako rezultat}
        pop:=stek[tekoven_broj_elementi];
        {i go namaluvame brojot na tekovni elementi(broevi) vo stekot}
        dec(tekoven_broj_elementi);          end
      else
        begin
          {ako stekot e prazen}
          writeln('Stekot e prazen !');
          {se vraka rezultat -1}
          pop:=-1;
        end;
      end;
end;

```

```

procedure pecati_stek;
var
  i:integer;
begin
  writeln('Elementite na stekot se, pocnuvajki od dnoto kon vrvot : ');
  for i:=1 to tekoven_broj_elementi do
    begin
      writeln(stek[i]);
    end;
end;

```

{Glavna programa}

```

begin
  inicijaliziraj_stek;

  push(12);
  push(23);
  push(9);

  pecati_stek;

  writeln('Izvadi element : ',pop());

  pecati_stek;

  writeln('Izvadi element : ',pop());

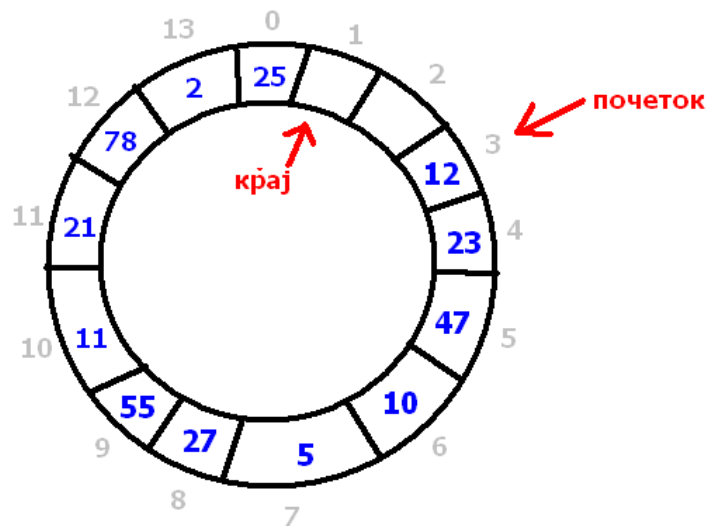
  pecati_stek;

  push(45);
  push(110);

  pecati_stek;

  readln;
end.

```

Пример програма за употреба на ред:

```

program primer_za_red;
var
  red:array[0..99]of integer;
  kapacitet:integer;
  tekoven_broj_elementi:integer;
  pocetok,kraj:integer;

procedure inicijaliziraj_red;
begin
  kapacitet:=100;
  tekoven_broj_elementi:=0;
  pocetok:=0;
  kraj:=0;
end;

procedure postavi_vo_red(x:integer);
begin
  {proveruvame dali ima se uste mesto vo redot}
  if tekoven_broj_elementi<kapacitet
  then
    begin
      inc(tekoven_broj_elementi);
      red[kraj]:=x;
      if kraj<kapacitet
      then
        inc(kraj)
      else
        kraj:=0;
      end
    end
  else
    writeln('Nema poveke mesto vo redot !');
  end;

function otstrani_prv():integer;
begin
  {proveruvame dali ima barem eden broj postaven vo redot}
  if tekoven_broj_elementi>0
  then
    begin
      dec(tekoven_broj_elementi);
    end
  end;
end;

```

```

        {go vrakame brojot od pocetokot na redot kako rezultat}
otstrani_prv:=red[pocetok];
if pocetok<kapacitet
    then
        inc(pocetok)
    else
        pocetok:=0;
    end
else
    begin
        {ako redot e prazen}
        writeln('Redot e prazen !');
        {se vraka rezultat -1}
        otstrani_prv:=-1;
    end;
end;

end;

procedure pecati_red;
var
    i:integer;
begin
    writeln('Elementite na redot se, pocnuvajki od prviot kon posledniot : ');
    if tekoven_broj_elementi>0
        then
            if (pocetok<kraj)
                then
                    begin
                        for i:=pocetok to kraj-1 do
                            writeln(red[i]);
                        end
                    else
                        begin
                            for i:=pocetok to kapacitet-1 do
                                writeln(red[i]);
                            for i:=0 to kraj-1 do
                                writeln(red[i]);
                            end;
                        end;
                    end;
end;

```

{Glavna programa}

```

begin
    inicijaliziraj_red;

    postavi_vo_red(12);
    postavi_vo_red(23);
    postavi_vo_red(45);

    pecati_red;

    writeln('Izvadi element : ',otstrani_prv());

    pecati_red;

    writeln('Izvadi element : ',otstrani_prv());

    pecati_red;

    postavi_vo_red(110);

    pecati_stek;

```

```
    readln;  
end.
```