

Rekurzivne funkcije

Rekurzivna funkcija je funkcija koja poziva sama sebe.

Neka je potrebno izračunati faktoriyel broja n . Faktoriyel broja n računa se po formuli:

$$n! = n * (n-1) * (n-2) * (n-3) * \dots * 2 * 1$$

Bez rekurzije, funkcija bi izgledala ovako:

```
main.cpp X
1  #include <iostream>
2  using namespace std;
3  int Faktoriyel(int broj)
4  {
5      int i, f=1;
6      for (i=1; i<=broj; i++)
7          f *= i;
8      return f;
9  }
10 int main()
11 {
12
13     cout << "Faktoriyel broja 5 je " << Faktoriyel(5) << endl;
14     return 0;
15 }
16
17
```

Faktoriyel broja 10 je 120

Process returned 0 (0x0) execution time : 0.016 s
Press any key to continue.

$$n! = n * (n-1) * (n-2) * (n-3) * \dots * 2 * 1$$

Računanje faktoriijela broja možemo posmatrati ovako:

$$n! = n * (n-1)!$$

$$(n-1)! = (n-1) * (n-2)!$$

$$(n-2)! = (n-2) * (n-3)!$$

$$(n-3)! = (n-2) * (n-3)! \dots 2! = 2 * 1!$$

.....

$$1! = 1$$

Kada računamo faktoriyel broja, množimo broj sa faktoriyelom broja minus 1. To se ponavlja sve dok ne računamo faktoriyel broja 1, koji iznosi 1. Da bismo izračunali faktoriyel broja n pozivamo računanje faktoriijela $(n-1)$, tj. funkcija poziva samu sebe. Ovaj postupak ponavljamo sve dok ne dođemo do neke konstantne vrijednosti koju funkcija vraća. Pošto funkcija unutar svog tijela poziva samu sebe, zovemo je rekurzivnom funkcijom.

Računanje faktoriijela broja na ovom principu realizuje slijedeće funkcija.

```

*main.cpp X
1  #include <iostream>
2  using namespace std;
3  int Faktorijel(int n)
4  {
5      if (n==1)
6          return 1;
7      else
8          return n * Faktorijel(n-1);
9  }
10 int main()
11 {
12
13     cout <<"Faktorijel broja 5 je "<< Faktorijel(5)<<endl;
14     return 0;
15 }
16
17

```

Kako bi bolje razumjeli kako radi rekurzivna funkcija, dodat ćemo par cout naredbi koje će ispisati kako se funkcija izvršava.

```

main.cpp X
1  #include <iostream>
2  using namespace std;
3  int Faktorijel(int n)
4  {
5      if (n==1){
6          cout <<"FaktorijelR(1) -- > 1"<<endl;
7          return 1;
8      }
9      else
10     {
11         cout <<"Pozivam --> "<< n <<" * Faktorijel("<< (n-1)<<" "<<endl;
12         return n * Faktorijel(n-1);
13     }
14 }
15 int main()
16 {
17
18     cout <<"Faktorijel broja 5 je "<<endl<< Faktorijel(5)<<endl;
19     return 0;
20 }
21

```

C:\Users\zahir\OneDrive\Desktop\CPP\ponavljanje\bin\Debug\ponav

```

Faktorijel broja 5 je
Pozivam --> 5 * Faktorijel(4)
Pozivam --> 4 * Faktorijel(3)
Pozivam --> 3 * Faktorijel(2)
Pozivam --> 2 * Faktorijel(1)
FaktorijelR(1) -- > 1
120
Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.

```

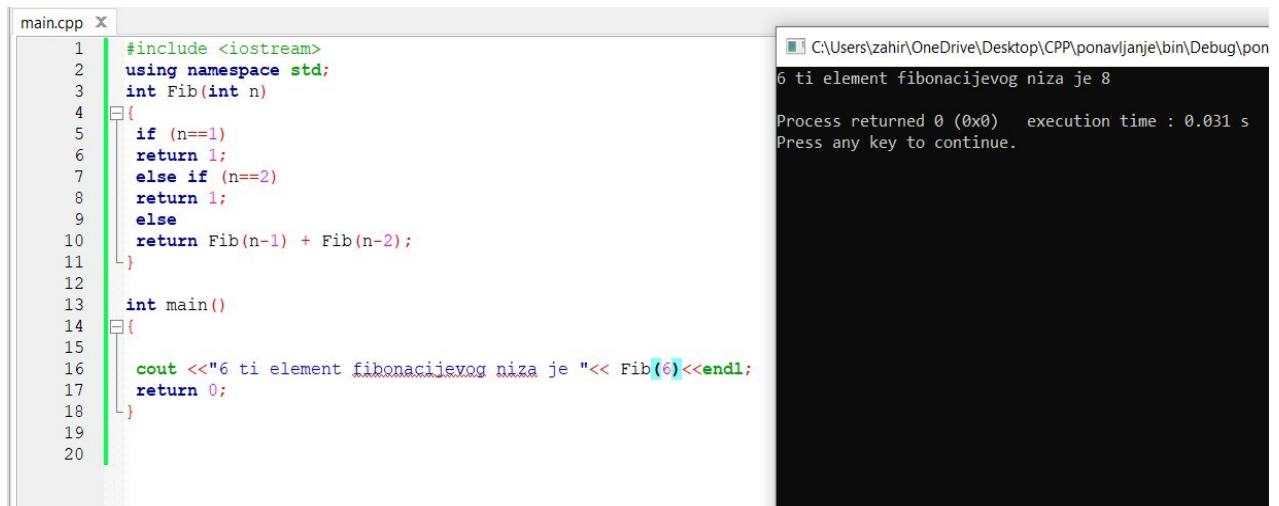
Ovo je bio jednostavan primjer rekurzivne funkcije koja unutar sebe ima samo jedan rekurzivan poziv. Ovakvu rekurzivnu funkciju nazivamo linearnom rekurzivnom funkcijom. Postoje funkcije koje unutar svog tijela imaju više od jednog poziva same sebe. Takve funkcije nazivamo nelinearnim. Kao primjer ove vrste funkcije imamo funkciju za računanje elemenata Fibonačijevog niza.

Fibonačijev niz čine brojevi: 1, 1, 2, 3, 5, 8, 13...

Prvi element niza ima vrijednost 1, drugi 1, a svi ostali mogu se izračunati kao zbir svoja dva prethodnika.

U opštem slučaju i-ti element niza može se izračunati po formuli: $f_i = f_{i-1} + f_{i-2}$

Funkcija koja vraća i-ti element Fibonačijevog niza može se veoma jednostavno riješiti rekurzivno, kako slijedi.



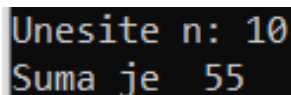
```
main.cpp X
1  #include <iostream>
2  using namespace std;
3  int Fib(int n)
4  {
5      if (n==1)
6          return 1;
7      else if (n==2)
8          return 1;
9      else
10         return Fib(n-1) + Fib(n-2);
11 }
12
13 int main()
14 {
15
16     cout <<"6 ti element fibonacijevog niza je "<< Fib(6)<<endl;
17     return 0;
18 }
19
20
```

C:\Users\zahir\OneDrive\Desktop\CPP\ponavljanje\bin\Debug\pon
6 ti element fibonacijevog niza je 8
Process returned 0 (0x0) execution time : 0.031 s
Press any key to continue.

Kada se računa vrijednost 6. Elementa niza, pozove se $\text{Fib}(5) + \text{Fib}(4)$.
Najprije se rješava poziv $\text{Fib}(5)$: $\text{Fib}(5) = \text{Fib}(4) + \text{Fib}(3)$.
Poziva se : $\text{Fib}(4) = \text{Fib}(3) + \text{Fib}(2)$.
Sada se poziva $\text{Fib}(3) = \text{Fib}(2) + \text{Fib}(1)$.

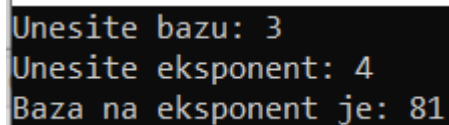
Zadaci za vježbu

1. Naći sumu brojeva od 1 do N. N unosi korisnik sa tastature. Zadatak riješiti upotrebom rekurzivne funkcije



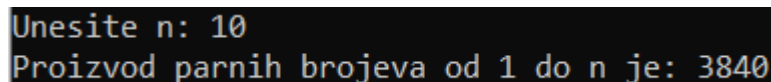
```
Unesite n: 10
Suma je 55
```

2. Napisati program koji koristeći funkciju Stepen preko dva parametra: baza i eksponent računa stepen baze na vrijednost eksponenta. Vrijednosti se unose sa tastature. Zadatak riješiti pomoću rekurzivne funkcije



```
Unesite bazu: 3
Unesite eksponent: 4
Baza na eksponent je: 81
```

3. Napisati rekurzivnu funkciju za izračunavanje proizvoda prvih n parnih prirodnih brojeva.



```
Unesite n: 10
Proizvod parnih brojeva od 1 do n je: 3840
```

4. Napisati rekurzivnu funkciju koja računa sumu cifara datog cijelog broja n

```
Unesite broj: 1284  
Suma cifara unesenog broja je: 15
```

5. Napisati rekurzivnu funkciju i za ispis sume vrijednosti prvih n brojeva niza:
 $1/2 + 1/3 + \dots + 1/n$.

```
Unesite broj: 5  
Suma niza je: 1.28333
```