



# **REKURZIVNE FUNKCIJE (REKURZIJE)**

# REKURZIVNE FUNKCIJE

- općenito rješavanje nekog većeg problema raščlanjujemo na manje zadatke koji se zatim povezuju u veće cjeline i tako dolazimo do konačnog rezultata
- ako se ulazni podaci mjere veličinom nekog parametra, često se rješenje pokušava naći tako da se krene od objekata manje veličine i zatim se postupno rješava probleme većih dimenzija
- zapravo rješavamo jednostavne probleme koji izgledaju isto kao početni i dalje ih pojednostavljujemo dok ne stignemo do trivijalnog problema kojem lako određujemo rješenje



## PRIMJER 1.

Neka je zadan prirodan broj  $N$ . Osmislimo rekurzivni način određivanja potencije  $2^N$ .

# OBJAŠNENJE

- $2^N$  možemo izračunati tako da  $N$  puta pomnožimo 2 sa samim sobom
- drugi način je da raščlanimo to na jednostavniji problem tj.  
 $2^N = 2 * 2^{(N-1)}$  na ovaj način imamo jednostavniji problem koji još možemo rastaviti na jednostavnije (rekurzija)
- kod rješavanja rekurzija najteži dio je odrediti pravilo (tj. relaciju i uvijet) rekurzije

# RJEŠENJE PRIMJERA 1. (BEZ REKURZIJE)

```
def potencija(n):  
    x = 1  
    for i in range(1, n+1):  
        x *= 2  
    return x  
n = int(input())  
print(potencija(n))
```

# RJEŠENJE PRIMJERA 1. (REKURZIVNA FUNKCIJA)

```
def potencija(n):  
    if n == 1:  
        return 2  
    else:  
        return 2*potencija(n-1)  
  
n = int(input())  
print(potencija(n))
```

# DEFINIRANJE REKURZIVNE FUNKCIJE

```
def ime_funkcije(parametri):  
    if uvjet_prekida:  
        return vrijednost  
    else:  
        return poziv_rekurzivne_relacije
```



## **PRIMJER 2.**

Napiši rekurzivnu funkciju za računanje zbroja prvih  $n$  prirodnih brojeva.



## RJEŠENJE PRIMJERA 2.

```
def zbroj(n) :  
    if n == 1:  
        return 1  
    else:  
        return n+zbroj(n-1)
```

```
n = int(input())  
print(zbroj(n))
```



## PRIMJER 3.

Napiši rekurzivnu funkciju koja će za cijeli broj  $a$  i potenciju  $n$  računati  $a^n$ .

## RJEŠENJE PRIMJERA 3.

```
def pot(a, n):  
    if n == 0:  
        return 1  
    else:  
        return a*pot(a, n-1)
```

```
a = int(input())  
n = int(input())  
print(pot(a, n))
```



## **PRIMJER 4.**

Napiši rekurzivnu funkciju koja će određivati duljinu zadanog stringa.

## RJEŠENJE PRIMJERA 4.

```
def dulj(s):  
    if s == "":  
        return 0  
    else:  
        return dulj(s[:-1]) + 1  
  
s = input()  
print(dulj(s))
```



## **PRIMJER 5.**

Napiši rekurzivnu funkciju za računanje n-tog Fibonaccijeva broja.

# RJEŠENJE PRIMJERA 5.

```
def Fibon(n):  
    if n == 1 or n == 2:  
        return 1  
    else:  
        return Fibon(n-1)+Fibon(n-2)  
  
n = int(input())  
print(Fibon(n))
```

## PRIMJER 6.

Napiši rekurzivnu funkciju za računanje najveće zajedničke mjere (najvećeg zajedničkog djeljitelja) pomoću Euklidova algoritma.

(Euklidov algoritam kaže da ako su dva broja  $a$  i  $b$  jednaka, onda je njihova zajednička mjera taj broj, a inače je jednaka mjeri brojeva  $|a-b|$  i najmanjem od broja  $a$  i  $b$ )



# RJEŠENJE PRIMJERA 6.

```
def NZM(a, b):  
    if a == b:  
        return a  
    elif a>b:  
        return NZM(a-b, b)  
    else:  
        return NZM(b-a, a)
```

```
a = int(input())  
b = int(input())  
print(NZM(a, b))
```

# ZADACI ZA VJEŽBU

1. Napiši rekurzivnu funkciju koja računa sljedeće zbrojeve:

a. 
$$\sum_{i=1}^n \frac{1}{i}$$

b. 
$$\sum_{i=1}^n (-1)^i i$$

2. Implementiraj jednu od inačica Euklidova algoritma:

- ako je  $a=b$ , onda je mjera brojeva  $a$  i  $b$  jednaka  $a$  ili  $b$
- ako je  $a < b$ , onda je mjera brojeva  $a$  i  $b$  jednaka mjeri brojeva  $b$  i  $a \bmod b$
- ako je  $b < a$ , onda je mjera brojeva  $a$  i  $b$  jednaka mjeri brojeva  $b$  i  $b \bmod a$

( $\bmod$  je ostatak pri djeljenju)

# ZADACI ZA VJEŽBU

- 3.** Napiši rekurzivni potprogram s jednim parametrom, prirodnim brojem  $n$ . Ako je paran,  $n$  postaje  $n/2$ , inače  $n$  postaje  $3n+1$ . Cilj je postići da  $n$  postane 1. Napišite funkciju koja će unositi prirodni broj  $n$  i vraćati koliko se puta  $n$  promijeni prije nego postane 1.
- 4.** Napiši rekurzivnu funkciju koja će računati minimum niza od  $n$  projeva.
- 5.** Napiši rekurzivnu funkciju koja će iz stringa izbacivati svaki drugi znak počevši od drugog znaka.
- 6.** Napiši rekurzivni potprogram koji će pretvarati broj iz baze 10 u bazu  $b$ .

# ZADACI ZA VJEŽBU

**7.** Napiši rekurzivnu funkciju koja će računati zbroj kvadrata prvih  $n$  prirodnih brojeva.

**8.** Napiši rekurzivnu funkciju koja će računati zbroj prvih  $n$  neparnih prirodnih brojeva.

**9.** Napiši rekurzivnu funkciju za računanje zbroja parnih brojeva u intervalu od  $a$  do  $b$ .

**10.** Napiši rekurzivnu funkciju za računanje  $n!$  ( $n$  faktorijela).

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 2 \cdot 1$$

**11.** Napiši rekurzivnu funkciju za računanje zbroja prvih  $n$  brojeva niza:

$$1/2 + 1/4 + \dots + 1/(2^n)$$

**12.** Napiši rekurzivnu funkciju koja će za učitano vrijednost  $n$  izračunati vrijednost izraza:  $1! + 2! + 3! + \dots + n!$

# ZADACI ZA VJEŽBU

- 13.** Napiši rekurzivnu funkciju koja provjerava je li broj prost ili složen.
- 14.** Napiši rekurzivnu funkciju koja računa broj znamenki zadanog prirodnog broja.
- 15.** Napiši rekurzivnu funkciju okreće znamenke zadanog prirodnog broja.
- 16.** Napiši rekurzivnu funkciju koja iz zadanog prirodnog broja izbacuje neparne znamenke
- 17.** Napiši rekurzivnu funkciju koja u zadanom prirodnom broju nakon svake neparne znamenke dodaje 0.
- 18.** Napiši rekurzivnu funkciju koja u zadanom prirodnom broju izbacuje svaku znamenku 3 ispred koje je znamenka 2.
- 19.** Napiši rekurzivnu funkciju koja u zadanom prirodnom broju određuje najmanju znamenku.
- 20.** Napiši rekurzivnu funkciju koja zadani dekadski prirodni broj pretvara u binarni.

# ZADACI ZA VJEŽBU

- 21.** Napiši rekurzivnu funkciju koja ispisuje članove niza  $F(i) = i*(i+1)/2$ .  
(1, 3, 6, 10...)
- 22.** Napiši rekurzivnu funkciju koja zbroj n članova niza  $F(i) = i*(i+1)/2$ .