

Termin XVII

8. februar 2020.

Redovni zadaci

1. rekurzija.py

Napisati rekurzivnu funkciju kojoj se prosleđuje prirodan broj. Pozivajući samu sebe, funkcija treba da ispiše sve prirodne brojeve manje od ili jednake tom broju počevši od broja 1.

PRIMER

ULAZ

4

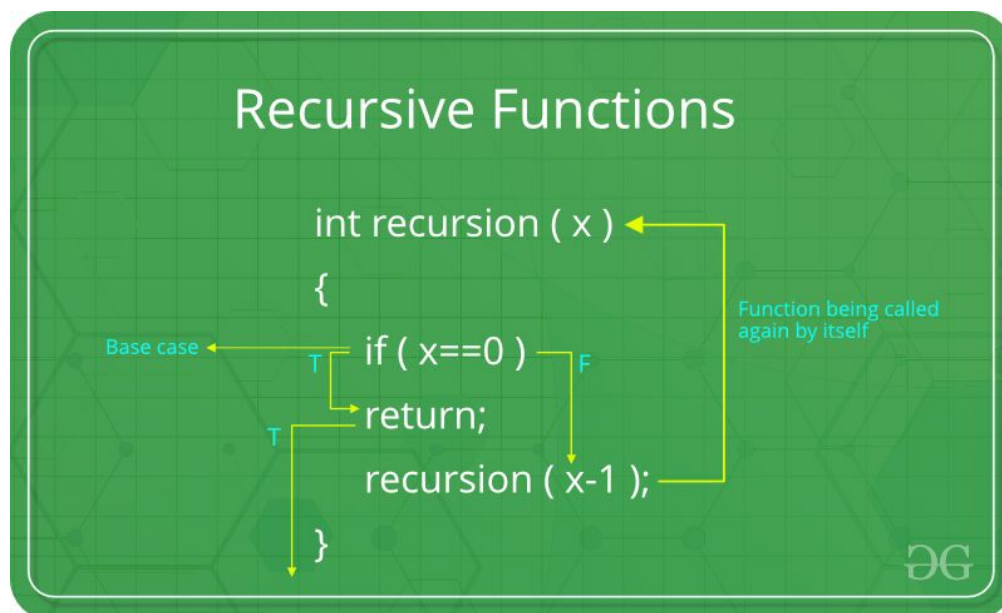
IZLAZ

1

2

3

4



2. faktorijel.py

Napisati rekurzivnu funkciju **faktorijel()** koja vraća ceo broj koji predstavlja faktorijel broja koji se prosleđuje kao argument. U jednoj liniji standardnog ulaza nalazi se prirodan broj n .

Napomena: Faktorijel broja n , u oznaci $n!$ računa se kao proizvod svih prirodnih brojeva od 1 do n (uključujući), odnosno: $n! = 1 * 2 * 3 * \dots * n$

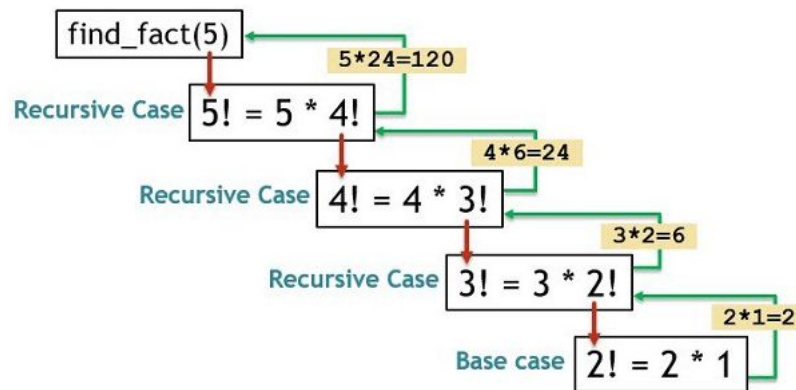
PRIMER

ULAZ

5

IZLAZ

120



3. fibonacijeva_sekvenca.py

Fibonačijeva sekvenca je sekvenca celih brojeva čija su prva dva člana 0 i 1, a svi ostali članovi dobijaju se dodavanjem prethodna dva broja. Na primer: 0, 1, 1, 2, 3, 5, 8, 13, itd.

Napisati rekurzivnu funkciju **fib_rek()** kojoj se prosleđuje ceo broj n , a vraća odgovarajuću vrednost koja predstavlja n -tu vrednost jednog člana Fibonačijeve sekvence. Od svih povratnih vrednosti funkcije, formirati listu i ispisati njene vrednosti.

PRIMER

ULAZ

7

IZLAZ

[0, 1, 1, 2, 3, 5, 8]

4. binarna_pretraga_rek.py

Napisati rekurzivni algoritam binarne pretrage koji nalazi zadati element u datoj listi. Ukoliko element nije nađen, vratiti vrednost -1. U glavnom delu programa, formirati proizvoljnu listu i uneti traženi broj x .

PRIMER

ULAZ

3 2 6 7 5 4
6

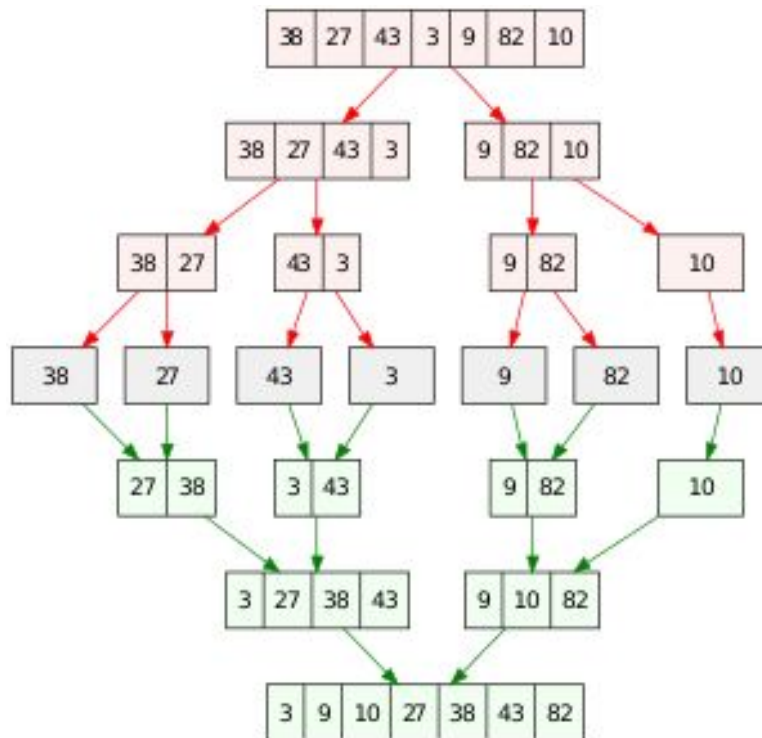
IZLAZ

2

5. uredjivanje_spajanjem.py

Napisati algoritam za **merge sort** (sortiranje umetanjem). U glavnom delu programa formirati neuređenu listu brojeva i sortirati pomoću **merge_sort()** funkcije, a zatim ispisati novouređenu listu.

Sortiranje umetanjem je algoritam koji je rekurzivan sam po sebi.



Dodatni zadaci

6. ispis_otpozadi.py

Napisati rekurzivnu funkciju koja vraća listu celih brojeva čiji su elementi uređeni u obrnutom poretku u odnosu na originalni. U glavnom delu programa, funkciji proslediti listu brojeva koja se učitava sa standardnog ulaza i odštampati povratnu vredost.

PRIMER

ULAZ

6 7 5 1 2 3 4

IZLAZ

4 3 2 1 5 7 6

7. broj_sadrzalaca.py

Napisati rekurzivnu funkciju koja određuje koliko se puta neki broj p pojavljuje u faktORIZACIJI nekog broja. U glavnom delu programa, uneti celobrojne vrednost x i p i pozvati rekurzivnu funkciju kojoj se prosleđuju ovi brojevi kao argumenti.

PRIMER

ULAZ

9
3

IZLAZ

2