

✓ 1. Napisati rekurzivnu funkciju koja sabira prvih N brojeva

```
def sum_first_n(n):
    if (n==0):
        return 0
    else:
        return n + sum_first_n(n-1)
```

```
print(sum_first_n(5))
```

↩ 15

✓ 2. Napisati rekurzivnu funkciju koja izracunava faktoriyel broja N

```
def fact_n(n):
    if (n==1):
        return 1
    else:
        return n*fact_n(n-1)
```

```
print(fact_n(5))
```

↩ 120

✓ 3. Napisati rekurzivnu funkciju koja stampa sve elemente liste redom

```
def print_elements(array):
    if (len(array)==0):
        return None
    else:
        print(array[0])
        return print_elements(array[1:])
```


```
arr = [1,2,3,4,5]
print_elements(arr)
```

↩ 1
2
3
4
5

✓ 5. Napisati rekurzivnu funkciju koja stampa string unazad

```
def inverse_string(string):
    if (len(string)==0):
        return ""
    else:
        return string[-1] + inverse_string(string[0:len(string)-1])
```

```
string = "Dejan"
print(inverse_string(string))
```

 najed

✓ 6. Napisati rekurzivnu funkciju koja stampa Nti broj Fibonaccijevog niza

```
def fibonacci_nth(n):
    if (n<2):
        return n
    else:
        return fibonacci_nth(n-2)+fibonacci_nth(n-1)
```


```
fibonacci_nth(3)
```

 2

✓ 7. Napisati rekurzivnu funkciju koja stampa N brojeva Fibonaccijevog niza

```
def fibonacci_n(n):
    if (n==0):
        return []
    elif (n==1):
        return 0
    elif (n==2):
        return [0,1]
    else:
        seq = fibonacci_n(n-1)
        print(seq)
        seq.append(seq[-1] + seq[-2])
        return seq
```

```
print(fibonacci_n(3))
```

 [0, 1]
[0, 1, 1]

✓ 9. Napisati rekurzivnu funkciju koja pronalazi najveći element niza

Double-click (or enter) to edit

```
def check_highest(arr):
    if (len(arr)==0):
        return None
    elif(len(arr)==1):
        return arr[0]
    else:
        if(arr[0]>arr[1]):
            arr.remove(arr[1])
        else:
            arr.remove(arr[0])
        return check_highest(arr)
```

```
arr = [1,23,4,5,324,1,4,5,10]
check_highest(arr)
```

↗ 324

```
def check_highest_(arr):
    if (len(arr)==0):
        return None
    elif(len(arr)==1):
        return arr[0]
    else:
        return max(arr[0],check_highest_(arr[1:]))
```

```
check_highest_(arr)
```

↗ 324

- ✓ 11. Napisati rekurzivnu funkciju sa dva parametra koja vraća stepenovan broj x.
Parametri su broj koji se stepenuje i stepen

```
def degree_num(x,deg):
    if (x==0):
        return 0
    elif(x==1):
        return 1
    elif(deg==0):
        return 1
    else:
        return x*degree_num(x,deg-1)
```

```
print(degree_num(1,0))
```

↗ 1