



# XVI susret

## String-ovi

`s = "Na danasnjoj radionici cemo obraditi lekciju vezanu za STRINGOVE."`

`s = input()` → ovako se učitava string

`duzinaStringa = len(s)`

`duzinaStringa = len('Danas je lep dan')`

Izraz	Rezultat
<code>'AB'+'cd'</code>	<code>'ABcd'</code>
<code>'A'+'7'+'B'</code>	<code>'A7B'</code>
<code>'Hi'*4</code>	<code>'HiHiHiHi'</code>



s='Python'

Naredba	Opis	Rezultat
s[0]	Prvo slovo u s	P
s[1]	Drugo slovo u s	y
s[-1]	Poslednje slovo u s	n
s[-2]	Preposlednje slovo u s	o

s = "Na danasnjoj radionici obraditi cemo lekciju vezanu za STRINGOVE."

```
for i in range(len(s)):
    print(s[i])
```

```
for i in range(len(s)):
    print(s[i], end="")
```





```
s='abcdefghij'
```

**Indeks:** 0 1 2 3 4 5 6 7 8 9

**Slova:** a b c d e f g h i j

Kod	Rezultat	Opis
s[2:5]	cde	Slova sa indeksom 2, 3, 4
s[:5]	abcde	Prvih pet slova
s[5:]	fghij	Slova od indeksa 5 do kraja
s[-2:]	ij	Poslednja dva slova
s[:]	abcdefghij	Ceo string
s[1:7:2]	bdf	Slova od indeksa 1 do 6, svako drugo
s[: :-1]	jihgfedcba	String unazad

```
s= 'Ana voli Milovana'
```

```
s=s[::-1]
```

```
print(s) → anavoliM ilov ana
```

```
print('Matematicka\nRadionica\n\nMladih')
```

```
→ Matematicka
```

```
→ Radionica
```

```
→
```

```
→ Mladih
```



Metod	Opis
lower()	Pretvara string u mala slova
upper()	Pretvara string u velika slova
replace(x,y)	Sve pojave x u stringu menja sa y
count(x)	Prebrojava koliko se puta x pojavljuje u stringu
index(x)	Vraća indeks pozicije prvog pojavljivanja x u stingu
isalpha()	Vraća true ako je svaki karakter u stringu alfa tipa

x, y → su stringovi, moraju biti ograničeni navodnicima, npr. 'abcd'...

s=':) ja VoLiM rAdiOnicU :)'

n=s.count(":)")

Metodi iz tabele pozivaju se

*promenljivaStringa.nazivMetoda(argumenti)*

s=s.lower()

print(" :) ima ", n) → :) ima 2

print(s) → :) ja volim radionicu :)



1. Tražite od korisnika da unese broj, a zatim štampajte po sledećoj šemi tako da poslednji broj bude onaj koji je korisnik uneo:

Ulaz: 4

Izlaz:

1

\*2

\*\*3

\*\*\*4

2. Napišite program koji traži od korisnika da unese string s pa konvertuje s u mala slova, da iz s ukloni sve tačke i zareze, i da štampa rezultat.
3. Ljudi često zaborave da zatvore zagradu u formulama. Napišite program koji traži od korisnika da unese formulu i štampa poruku da li formula ima isti broj levih i desnih zagrada.
4. Jedan jednostavan način da se proceni broj reči u stringu je da se prebroji broj blanko znakova (space) u stringu. Napišite program koji traži od korisnika da unese tekst, a onda mu vraća procenjeni broj reči u tom stringu.



5. Napišite program koji traži od korisnika da unese dva stringa iste dužine. Program treba da proveri da li su stvarno uneti stringovi iste dužine. Ako nisu, program štampa odgovarajuću poruku i završava rad. Ako jesu iste dužine program treba da štampa alternativno slova iz oba stringa.

Na primer, ako je korisnik uneo stringove **abcde** i **ABCDE**

Program će štampati **AaBbCcDdEe**

6. Bez korišćenja **in** operatora, napišite program koji traži od korisnika da unese string i jedna znak, pa da štampa da li se taj znak pojavljuje u stringu.

Napomena:

Ako poredite **aa** i **bb** npr **aa == bb**  
**aa** i **bb** moraju biti istog tipa

**"100"** nije isto **100** !!!

7. Potrebno je dešifrovati poruku 123121452. Ako znamo:

4 – u

5 – r

1 – t

2 – a

3 – s