

Praktikum iz programiranja 1



2024/25



String funkcije

- String u stringu – find, index

`string.find(pattern[,begin[,end]])`

```
str1 = "Ovo je reprezentacija nekog stringa!"  
str2 = "taci"  
print (str1.find(str2))           15  
print (str1.find(str2, 10))       15  
print (str1.find(str2, 16))       -1  
print (str1.find(str2, 10, 20))   15  
print (str1.find(str2, 10, 16))   -1
```

String funkcije

- String u stringu – **find, index**

`string.index(pattern[,begin[,end]])`

```
str1 = "Ovo je reprezentacija nekog stringa!"
```

```
str2 = "taci"
```

```
print (str1.index(str2))
```

```
print (str1.index(str2, 16))
```

15

ValueError: substring not found

String funkcije

- Broj ponavljanja stringa u stringu – **count**
- Konvertovanje svih slova u velika, odnosno mala – **upper, lower**
- Promena dela stringa novim stringoma – **replace**

```
s = 'Ovaj predmet se zove PRAKTIKUM'
```

```
print (s.count(' '))
```

6

```
print (s.upper())
```

OVAJ PREDMET SE ZOVE PRAKTIKUM

```
print (s.lower())
```

ovaj predmet se zove praktikum

```
print (s.replace('a', 'A'))
```

OvAj predmet se zove PrAktikum

Stringovi su nepromenljivi

- Posmatrajmo sledeći deo koda:

```
rec = "Osnovi programiranja"  
rec[0] = 'A'  
print(rec)
```

- Umesto očekivanog izlaza:
Asnovi programiranja
- Izlaz je runtime error:
TypeError: 'str' object does not support item assignment

- Predloženo rešenje – formiranje nove reči na osnovu stare:

```
nova_rec = 'A'+rec[1:]  
print(nova_rec)
```

Poređenje stringova

```
ime = input()
```

```
if ime=="Marko":  
    print("Da, moje ime je Marko")
```

ili svi slučajevi:

```
if ime < "Marko":  
    print("Uneto ime je ispred Marka")  
elif ime > "Marko":  
    print("Uneto ime je iza imena Marko")  
else:  
    print("Uneto ime je Marko")
```

Primer

- Šta je rezultat rada sledećeg programa?

```
prvi_sabirak = input("Unesi prvi sabirak: ")  
drugi_sabirak = input("Unesi drugi sabirak: ")  
zbir = prvi_sabirak + drugi_sabirak  
print(zbir)
```

Primer

- Tekst je palindrom ako se čita sdesna nalevo isto kao i sleva nadesno (radar, kajak). Napisati funkciju koja testira da li je neki tekst palindrom. Problem rešiti kako iterativno, tako i rekurzivno.

```
def palindromI(tekst):  
    n = len(tekst)  
    limit = n//2  
    for i in range(limit):  
        if tekst[i] != tekst[-i-1]:  
            return False  
    return True
```


Primer

- Napisati program kojim se određeni tekst menja
 - tako što se sva pojavljivanja karaktera „A“ menjaju stringom „XXX“.
 - tako što se iza svakog karaktera „A“ dodaje string „XXX“.
 - tako što se iza poslednjeg karaktera „A“ dodaje string „XXX“.

Primer

```
s=input()  
#s="dAnnas je DAN za rAdost"  
  
s=s.replace('A','XXX')  
print(s)
```

Primer

```
br=0
while s.find('A',br)!=-1:
    br=s.find('A',br)
    s=s[:br+1]+"XXX"+s[br+1:]
    print(s)
    br=br+3
```

```
s=s.replace('A','A'+'XXX')
print(s)
```

Primer

```
x=s  
br=0  
while x.find('A',br)!=-1:  
    br=x.find('A',br)  
    s=x[:br+1]+"XXX"+x[br+1:]  
    br=br+1  
  
print(s)
```

Primer

- Napisati program kojim se određuje tekst koji se dobija brisanjem iz datog teksta svih pojavljivanja date reči. Taj postupak se ponavlja sve dok se tekst njime menja.

```
s1="babrarkadabrabbrr"
```

```
s2="br"
```

```
n=len(s2)
```

```
while s1.find(s2)!=-1:
```

```
    br=s1.find(s2)
```

```
    s1=s1[:br]+s1[br+n:]
```

```
    print(s1)
```

```
print(s1)
```