

All Code Clubs must be registered. By registering your club we can measure our impact, and we can continue to provide free resources that help children learn to code. You can register your club at codeclubworld.org.

Uvod:

U ovom projektu naučit ćeš kako napraviti vlastiti program za šifriranje te poslati tajne poruke prijatelju i primiti tajne poruke prijatelja.

```
Unesite poruku:  pozdrav!  
Unesite zeljeni kljuc (1-26):  5  
Tvoja nova poruka je uteiwfa!
```

```
Unesite poruku:  uteiwfa!  
Unesite zeljeni kljuc (1-26):  -5  
Tvoja nova poruka je pozdrav!
```



Activity Checklist

Follow these **INSTRUCTIONS** one by one



Test your Project

Click on the green flag to **TEST** your code



Save your Project

Make sure to **SAVE** your work now

Korak 1: Cezarova šifra

Šifra je vrsta tajnog kôda gdje se slova međusobno zamjenjuju tako da nitko ne može pročitati tvoju poruku.

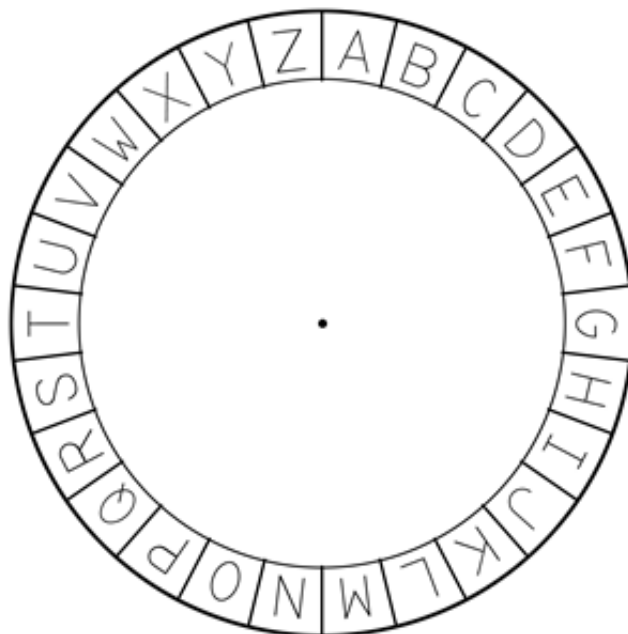
Koristit ćeš jedan od najstarijih načina šifriranja, Cezarovu šifru, koji je nazvan po Juliju Cezaru.

Prije nego započnemo sa kodiranjem, pokušajmo sakriti riječ koristeći Cezarovu šifru.

Zadatci

- Skrivanje riječi naziva se enkripcija. ☐

Započnimo sa šifriranjem slova 'a'. Nacrtat ćemo englesku abecedu u krugu kako bi sakrio slovo:



- Da bi od običnog slova napravio tajno šifrirano slovo, moraš imati tajni ključ. Koristit ćemo broj 3 kao tajni ključ (ali možeš koristiti bilo koji broj). ☐

Za enkripciju slova 'a', pomakni se za 3 slova unaprijed, u

smjeru kazaljke na satu. Tako ćeš dobiti slovo 'd':



- Iskoristi naučeno i šifriraj riječ. Primjerice, kada šifriramo riječ 'hello' dobit ćemo 'khood'. Pokušaj!



- $h + 3 = k$
- $e + 3 = h$
- $l + 3 = o$
- $l + 3 = o$
- $o + 3 = r$

- Vraćanje teksta u početni oblik naziva se dekripcija. Za dešifriranje riječi jednostavno oduzmi tajni ključ od slova umjesto da ga dodaš:



- $k - 3 = h$
- $h - 3 = e$
- $o - 3 = l$
- $o - 3 = l$
- $r - 3 = o$

Izazov: Isprobaj Cezarovu šifru

Možeš li poslati tajnu riječ prijatelju? Prvo se morate dogovoriti što će biti tajni ključ.

Možete razmijeniti i cijele rečenice!

Korak 2: Šifriranje slova

Napišimo program u Pythonu koji će šifrirati znakove.

Zadatci

- Otvori prazan Python predložak u Trinketu: jump.to/cc/python-new. Ako čitaš ovo online, možeš koristiti i ugrađenu verziju ovog trinketa koja se nalazi ispod. ☐

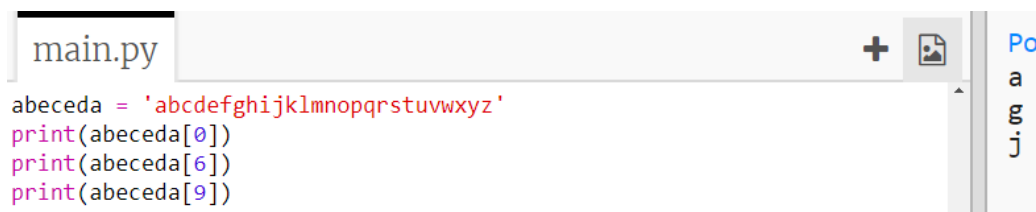
- Umjesto crtanja abecede u krugu, definirajmo ju kao varijablu `abeceda`. ☐

```
abeceda = 'abcdefghijklmnopqrstuvwxyz'
```

- Svako slovo abecede ima svoju poziciju, počevši od 0. Dakle, slovo 'a' je na nultoj poziciji abecede, a 'c' na poziciji 2. ☐

```
abeceda = 'a0b1c2d3e4
```

- Za ispis slova koje se nalazi u varijabli `abeceda`, upiši poziciju tog slova unutar uglatih zagrada. ☐



```
main.py
abeceda = 'abcdefghijklmnopqrstuvwxyz'
print(abeceda[0])
print(abeceda[6])
print(abeceda[9])
```

Možeš izbrisati `print` naredbe nakon što si testirao kôd.

- Zatim moraš spremiti tajni `ključ` u varijablu. ☐

```
abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3
```

- Sada pitaj korisnika da unese jedno slovo (odnosno `znak`) koje će se šifrirati. ☐

```

abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3

znak = input('Unesi jedan znak: ')

```

- Pronađi `poziciju` tog `znaka`.



```

abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3

znak = input('Unesi jedan znak: ')

pozicija = abeceda.find(znak)

```

- Ispiši spremljenu `poziciju` kako bi ju testirao. Primjerice, testiraj je li znak 'e' na poziciji 4 u abecedi.



The screenshot shows a Trinket.io code editor with a file named 'main.py'. The code in the editor is:

```

abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3

znak = input('Unesi jedan znak: ')

pozicija = abeceda.find(znak)
print(pozicija)

```

The output on the right shows the prompt 'Unesi jedan znak: ' followed by the input 'e' and the result '4'.

- Dodaj `ključ` `poziciji` za šifriranje `znaka`.



```

abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3

znak = input('Unesi jedan znak: ')

pozicija = abeceda.find(znak)
print(pozicija)

novaPozicija = pozicija + kljuc

```

- Testiraj svoj novi kôd. S obzirom da je tvoj `ključ` broj 3, `poziciji` bi se trebao dodati broj 3, a zatim se spremi u varijablu `novaPozicija`.



Primjerice, slovo 'e' je na poziciji 4. Kako bi se slovo šifriralo, poziciji se dodaje ključ (3) i naposljetku dobije 7.

```
main.py
abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3

znak = input('Unesi jedan znak: ')

pozicija = abeceda.find(znak)
print(pozicija)

novaPozicija = pozicija + kljuc
print(novaPozicija)
```

Powered by trinket
Unesi jedan znak: e
4
7

- Što se dogodi kada pokušaš šifrirati slovo 'y'?



```
Unesi jedan znak: y
24
27
```

Primjećuješ li da je novaPozicija 27, a u engleskoj abecedi nema 27 slova!

- Ako upišeš % nova pozicija će se, nakon što dođe do pozicije 26, vratiti na poziciju 0.



```
main.py
abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3

znak = input('Unesi jedan znak: ')

pozicija = abeceda.find(znak)
print(pozicija)

novaPozicija = (pozicija + kljuc) % 26
print(novaPozicija)
```

Powered by trinket
Unesi jedan znak: y
24
1

- Konačno, sada ćeš ispisati slovo koje se nalazi na novoj poziciji.



Primjerice, kada dodamo ključ slovu 'e' dobit ćemo 7, a slovo koje se nalazi na poziciji 7 u abecedi je 'h'.

```
main.py
abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3

znak = input('Unesi jedan znak: ')

pozicija = abeceda.find(znak)
print(pozicija)

novaPozicija = (pozicija + kljuc) % 26
print(novaPozicija)

noviZnak = abeceda[novaPozicija]
print(noviZnak)
```

Powered by trinket
Unesi jedan znak: e
4
7
h

- Isprobaj svoj kôd. Možeš i ukloniti neke od print naredbi i ispisati samo novi znak na kraju.



```
main.py
abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3

znak = input('Unesi jedan znak: ')

pozicija = abeceda.find(znak)

novaPozicija = (pozicija + kljuc) % 26

noviZnak = abeceda[novaPozicija]
print('Novi znak je:', noviZnak)
```

Powered by trinket
Unesi jedan znak: e
Novi znak je: h



Spremi projekt

Izazov: Promjenjivi ključevi

Izmijeni svoj program tako da omogućava korisniku unošenje proizvoljnog ključa. Korisnikov unos spremi u varijablu `ključ`.

Ne zaboravi koristiti funkciju `int()` za pretvaranje unosa u cijeli broj.

Možeš koristiti i negativni ključ za dešifriranje poruke!



Spremi projekt

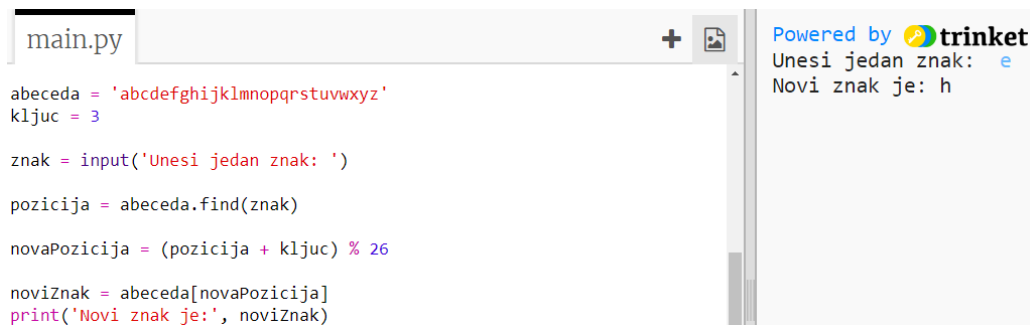
Korak 2: Šifriranje poruka

7

Podesimo program da šifrira cijele poruke, umjesto da šifrira i dešifrira jedan po jedan znak!


Zadatci

- Prvo provjeri izgleda li tvoj kôd ovako: ☐



```
main.py
abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3

znak = input('Unesi jedan znak: ')
pozicija = abeceda.find(znak)
novaPozicija = (pozicija + kljuc) % 26
noviZnak = abeceda[novaPozicija]
print('Novi znak je:', noviZnak)
```

Powered by  trinket
Unesi jedan znak: e
Novi znak je: h

- Definiraj varijablu u koju ćeš spremiti svoju novu šifriranu poruku. ☐

```
abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3
novaPoruka = ''

znak = input('Unesi jedan znak: ')

pozicija = abeceda.find(znak)

novaPozicija = (pozicija + kljuc) % 26

noviZnak = abeceda[novaPozicija]
print('Novi znak je:', noviZnak)
```

- Izmijeni kôd tako da sprema korisnikovu poruku, a ne samo jedan znak. ☐


```
abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3
novaPoruka = ''
```

```
poruka = input('Unesite poruku: ')
```

```
znak = input('Unesi jedan znak: ')
```

```
pozicija = abeceda.find(znak)
```

```
novaPozicija = (pozicija + kljuc) % 26
```

```
noviZnak = abeceda[novaPozicija]
print('Novi znak je:', noviZnak)
```

- Svom kôdu dodaj `for` petlju, a zatim uvuci ostatak kôda kako bi se ponavljao za svaki znak poruke. ☐

```
abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3
novaPoruka = ''
```

```
poruka = input('Unesite poruku: ')
```

```
for znak in poruka:
    pozicija = abeceda.find(znak)
    novaPozicija = (pozicija + kljuc) % 26
    noviZnak = abeceda[novaPozicija]
    print('Novi znak je:', noviZnak)
```

- Testiraj kôd. Svaki znak poruke trebao bi se šifrirati i ispisivati jedan po jedan. ☐

```
main.py
#!/bin/python3

abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3
novaPoruka = ''

poruka = input('Unesite poruku: ')

for znak in poruka:
    pozicija = abeceda.find(znak)
    novaPozicija = (pozicija + kljuc) % 26
    noviZnak = abeceda[novaPozicija]
    print('Novi znak je:', noviZnak)
```

Powered by trinket

Unesite poruku: pozdrav

Novi znak je: s

Novi znak je: r

Novi znak je: c

Novi znak je: g

Novi znak je: u

Novi znak je: d

Novi znak je: y

- Dodajmo svaki šifrirani znak u varijablu `novaPoruka`.



```
abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3
novaPoruka = ''

poruka = input('Unesite poruku: ')

for znak in poruka:
    pozicija = abeceda.find(znak)
    novaPozicija = (pozicija + kljuc) % 26
    noviZnak = abeceda[novaPozicija]
    print('Novi znak je:', noviZnak)
    novaPoruka += noviZnak
```

- Naredbom `print` možeš ispisati varijablu `novaPoruka` dok se šifrira.



```
main.py
#!/bin/python3

abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3
novaPoruka = ''

poruka = input('Unesite poruku: ')

for znak in poruka:
    pozicija = abeceda.find(znak)
    novaPozicija = (pozicija + kljuc) % 26
    noviZnak = abeceda[novaPozicija]
    print('Novi znak je:', noviZnak)
    novaPoruka += noviZnak
print(novaPoruka)
```

Powered by trinket

Unesite poruku: pozdrav

Novi znak je: s

s

Novi znak je: r

sr

Novi znak je: c

src

Novi znak je: g

srcg

Novi znak je: u

srcgu

Novi znak je: d

srcgud

Novi znak je: y

srcgudy

- Izbrišeš li prazna mjesta prije naredbe `print` šifrirana poruka će se ispisati samo jednom na kraju. Možeš izbrisati i dio kôda koji ispisuje pozicije znakova.




```
main.py
#!/bin/python3

abeceda = 'abcdefghijklmnopqrstuvwxyz'
kljuc = 3
novaPoruka = ''

poruka = input('Unesite poruku: ')

for znak in poruka:
    pozicija = abeceda.find(znak)
    novaPozicija = (pozicija + kljuc) % 26
    noviZnak = abeceda[novaPozicija]
    #print('Novi znak je:', noviZnak)
    novaPoruka += noviZnak

print('Tvoja nova poruka je', novaPoruka)
```

Powered by  trinket
Unesite poruku: pozdrav
Tvoja nova poruka je srcgudy



Spremi projekt

Korak 3: Dodatni znakovi

Neki se znakovi ne nalaze u abecedi, zbog čega se javljaju greške.



Zadatci

- Isprobaj kako radi tvoj kôd kada koristiš neke znakove koji se ne nalaze u abecedi.



Primjerice, možeš napisati poruku `hej bok!!`.

```
Unesite poruku:  hej bok!!
Tvoja nova poruka je khmcerncc
```

Primijeti da su razmak i znak uskličnika `!` šifrirani kao slovo `'c'!`

- Da bi ovo popravio, program mora prevoditi znak samo ako se nalazi u abecedi. To ćeš napraviti dodavanjem `if` naredbe u kôd i uvlačenjem ostatka kôda.



```
for znak in poruka:
    if znak in abeceda:
        pozicija = abeceda.find(znak)
        novaPozicija = (pozicija + kljuc) % 26
        noviZnak = abeceda[novaPozicija]
        novaPoruka += noviZnak
```

- Testiraj kôd koristeći istu poruku kao prije. Što se ovaj put događa? ☐

Unesite poruku: **hej bok!!**
Tvoja nova poruka je khmern

Sada tvoj kôd preskače svaki znak koji nije u abecedi.

- Najbolje bi bilo kada tvoj kôd ne bi šifrirao ništa što nije u abecedi, nego jednostavno koristio originalni znak. ☐

Kôdu dodaj `else` naredbu koja će dodati originalni znak šifriranoj poruci.

```
for znak in poruka:
    if znak in abeceda:
        pozicija = abeceda.find(znak)
        novaPozicija = (pozicija + kljuc) % 26
        noviZnak = abeceda[novaPozicija]
        novaPoruka += noviZnak
    else:
        novaPoruka += znak
```

- Testiraj kôd. Svi znakovi abecede trebali bi biti šifrirani, dok su svi ostali znakovi u originalnom obliku! ☐

Unesite poruku: **hej bok!!**
Tvoja nova poruka je khm ern!!



Spremi projekt

Izazov: Šifriranje i dešifriranje poruka

Šifriraj nekoliko poruka te ih, zajedno sa tajnim ključem, daj prijatelju. Neka ih tvoj prijatelj pokuša dešifrirati koristeći svoj program!

Možeš napraviti i kopiju projekta te napraviti poseban program za dešifriranje poruka.



Spremi projekt

Izazov: Kalkulator prijateljstva

Napiši program koji će, računanjem bodova prijateljstva, pokazivati koliko dobro si dvije osobe odgovaraju.

```
Unesi imena dviju osoba: Iva & Petra
Vas rezultat prijateljstva je : 35
```

Program neka prođe kroz svaki znak koji se nalazi u imenima dvaju osoba. Kada pronađe slovo koje nosi određen broj bodova, neka te bodove doda u varijablu `rezultat`.

Odredi kakva će biti pravila za davanje bodova. Primjerice, možeš davati bodove za samoglasnike ili za znakove koji se nalaze u riječi “prijatelj”:

```
if znak in 'aeiou':
    rezultat += 5
if znak in 'prijatelj':
    rezultat += 10
```

Korisnicima možeš prikazati i posebnu poruku na osnovi njihovog rezultata.

```
if rezultat > 100:
    print('Najbolji prijatelji!')
```



Spremi projekt