4. Laboratorijska vaja

Cezarjeva koda

Šifriranje (kodiranje) sporočil je ena najstarejših oblik ohranjanja skrivnosti. Je smrtno resna zadeva, ki ohranja pri življenju vojake, ohranja skrivne pogovore med voditelji in nenazadnje, ohranja zasebnost vseh naših sporočil, ki jih pošiljamo po medmrežju.

Tudi Julij Cezar se je zavedal pomembnosti skrivnih sporočil in si je za kodiranje svojih ukazov, ki jih je pošiljal častnikom, zamislil naslednji postopek. Vsako od črk v svojem sporočilu je zamaknil za tri mesta navzgor po abecedi; črko a je tako v kodiranem sporočilu predstavil s č-jem, črko b je predslavil z d-jem, in tako dalje. Na koncu abecede je uporabil ovijanje, tako da je na primer črko z predstavil z b-jem, črko ž pa s c-jem. Cezarjevi častniki (in nihče drug) so vedeli, da je za dekodiranje prejetega sporočila potreben zamik za tri znake abecede v levo, da so sporočilo lahko razumeli.

Primer ukaza in njegove kodirane različice (z zamikom 3 v desno):
primiteobeliksa
štlplzhsdholnuč

V naši vaji bomo uporabili slovensko abecedo (25 črk), z dodanim podčrtajem kot 26. znak:
abcčdefghijklmnoprsštuvzž_
čdefghijklmnoprsštuvzž_abc (koda z zamikom za tri mesta v desno)

Besedilo naloge

Napiši program JavaScript, ki kodira sporočilo, podano v obliki tabele znakov. Program naj vsebuje podprogram za kodiranje kot tudi za dekodiranje sporočila s poljubnim zamikom.

Pomoč

Za pomoč podajamo tabelo znakov (`CHARTABLE`), ki jo uporabite kot osnovo za kodiranje in dekodiranje. Podano je tudi sporočilo (`PLAIN_MESSAGE`), ki ga uporabite za preizkušanje kode. Zadnji izmed elementov je 0 – čuvaj, ki označuje konec tabele znakov.

Opomba: v programu se izognite uporabi postopka `length()` jezika JavaScript.

```
CHARTABLE = [     'a', 'b', 'c', 'č', 'd',
                  'e', 'f', 'g', 'h', 'i',
                  'j', 'k', 'l', 'm', 'n',
                  'o', 'p', 'r', 's', 'š',
                  't', 'u', 'v', 'z', 'ž','_', 0];

PLAIN_MESSAGE = ['p','r','e','m','i','k','_',
                  'v','s','e','h','_',
                  'e','n','o','t','_',
                  'i','z','_',
                  'b','r','e','t','a','n','i','j','e','_',
                  'v','_',
                  'g','a','l','i','j','o','_',
                  't','a','k','o','j','_',
                  'c','a','e','s','a','r',0];
```

Predlagamo, da sprva napišete podprogram, ki sprejme poljuben znak abecede in vrne indeks tega znaka v tabeli. Primer:

```
-> getCharIndex('a')
<- 0
-> getCharIndex('p')
<- 16
```

Nato ustvarite podprogram, ki zakodira en sam znak z določenim zamikom. Primer:

```
-> encryptCharCaesar('a', 3)
<- 'č'
-> encryptCharCaesar('ž', 3)
<- 'b'
```

Sedaj z uporabo zanke preberite besedilo in vsako črko nadomestite s kodirano različico.

Ko imate sporočilo zakodirano, potrebujete še podprogram za njegovo dekodiranje. Primer:

```
-> decryptCharCaesar('č', 3)
<- 'a'
-> decryptCharCaesar('b', 3)
<- 'ž'
```

Kakšno pomembno informacijo so poslali legionarji s terena Juliusu Cezarju (zamik 22)?

```
IMPORTANT_MESSAGE = [
     'v','n','p','b','m','e',
     'g','n','u','l','m','e',
     'f','b','p','u','k','z',
     'b','h','e','g','n','u',
     'l','k','z','b','č','j',
     'e','h', 0];
```

Dodatek (Vigenèrjeva koda) – za pogumne

Cezarjeva kodo je lahko razbiti, tudi če ne poznate zamika v kodiranju (preiskusite vse možne zamike, ne obstaja jih več kot je dolžina dogovorjene abecede, torej 26).

Blaise de Vigenère je zato leta 1585 predlagal izboljšano verzijo Cezarjeve kode, ki temelji na vnaprej dogovorjenem geslu. Vsak znak gesla predstavlja različen zamik po abecedi. Naslednje geslo predstavlja sledeče zamike za vsak znak sporočila:

a	**m**	**f**	**o**	**r**	**a**
0	**13**	**6**	**15**	**17**	**0**

Sporočilo se zakodira z geslom na sledeči način:
Sporočilo: **primiteobeliksa**
Geslo: **amforaamforaamf**
Kodirano: **pdocatecgtčikef**