

## Вежбе 5

### ## 1

## Одштампати првих n Unicode карактера, заједно са UCP бројем.

```
##n = int(input('n = '))
##for i in range(n+1):
##    print(i,chr(i))
```

### ## 2

## Ana i Marko dogovorili su se da razmenjuju  
## šifrovane poruke na sledeci način: jedinstveni Unicode broj  
## svakog karaktera iz originalne  
## poruke, treba pomnožiti sa a, i na rezultat dodati broj b.  
## Potom se broj prevodi  
## u odgovarajući karakter šifrovane poruke.  
## Napisati program koji će, za uneto a i b,  
## šifrovati (dešifrovati) ulaznu poruku. Ako je a = 2 i b = 3,  
## dešifrovati sledeću Aninu poruku Marku: “iÖĚŎŸáCéÍcĚŘÁéCíCm”.

```
##def sifrira(poruka, mod, a, b):
##    ## mod moze da bude s ili d
##    n = len(poruka)
##
##    if mod == 's':
##        for i in range(n):
##            print(chr(a*ord(poruka[i]) + b),end='')
##
##    elif mod == 'd':
##        for i in range(n):
##            print(chr( (ord(poruka[i])-b)//a),end='')
##    else:
##        print('greska u unosu')
##def sifrira(poruka, mod, a, b):
##    ## mod moze da bude s ili d
##    rezultat = 'test'
##
##    if mod == 's':
##        for k in poruka:
##            rezultat += chr(a*ord(k) + b)
##    elif mod == 'd':
##        for k in poruka:
##            rezultat += chr((ord(k)-b)//a)
##    else:
##        print('greska u unosu')
##
##    print(rezultat)
```

### ## 3

## Написати функцију која за целобројну листу  
## враћа средњу вредност и највећи број.

```
##def sr_max(lista):  
##  
##    n = len(lista) # duzina kolekcije  
##    s = 0         # za sada je suma jednaka 0  
##    max_el = lista[0] # za sada nam je najveći el prvi u listi  
##  
##    for e in lista: # po svim elementima u listi  
##        s += e      # dodaj el u sumu  
##        if e > max_el: # da li je tekuci el veci od starog max  
##            max_el = e # ako jeste, azuriraj i stavi da je max e  
##  
##    return [s/n, max_el]  
##  
##  
##n = int(input('duzina liste'))  
##  
##lis = []  
##  
##for i in range(n):  
##    lis.append(int(input('el = ')))  
##  
##rezultat = sr_max(lis)  
##  
##print('srednja vrednost je: ', rezultat[0])  
##print('najveća vrednost je: ', rezultat[1])
```

### ## 4

## Написати функције за пресек и унију елемената листе.

```
##def unija(l1, l2):  
##  
##    lu = []  
##  
##    for el in l1:  
##        if el not in lu:  
##            lu.append(el)  
##  
##    for el in l2:  
##        if el not in lu:  
##            lu.append(el)  
##  
##    return lu  
##  
##def presek(l1, l2):  
##  
##    lp = []  
##  
##    for el in l1:  
##        if el in l2 and el not in lp:  
##            lp.append(el)  
##  
##    return lp
```

```
## 5
```

```
## Израчунати Jaccard индекс сличности за две листе речи.
```

```
##tekst1, tekst2 = input('unesi tekst 1: '), input('unesi tekst 2: ')
```

```
##
```

```
##reci1, reci2 = tekst1.split(), tekst2.split()
```

```
##
```

```
##print('slicnost je',len(presek(reci1, reci2))/len(unija(reci1,reci2)))
```

```
## 6
```

```
## Написати функцију којом се сабирају две квадратне матрице.
```

```
##def sab_mat(a,b):
```

```
##     c=[]
```

```
##     for i in range(len(a)):
```

```
##         red=[]
```

```
##         for j in range(len(a[0])):
```

```
##             red.append(a[i][j] + b[i][j])
```

```
##         c.append(red)
```

```
##     for i in range(len(c)):
```

```
##         print(c[i])
```