In [1]:

```
f = open('datoteke/teze.txt')
najvecja_vrednost = 0
for vrstica in f:
    trenutna_vrednost = int(vrstica)
    if trenutna_vrednost > najvecja_vrednost:
        najvecja_vrednost = trenutna_vrednost
print(najvecja_vrednost)
```

```
85
```

In [2]:

```
imena = ['Ana', 'Berta', 'Cilka', 'Dani']
```

In [3]:

```
len(imena)
```

Out[3]:

```
4
```

In [4]:

```
teze = [56, 76, 80, 67, 60]
```

In [7]:

```
vsota = 0
for m in teze:
    vsota = vsota + m
print(vsota), print(len(teze))
povp = vsota / len(teze)
print(povp)
```

```
339
5
67.8
```

In [8]:

```
sum(teze)/len(teze)
```

Out[8]:

```
67.8
```

In [9]:

```
sum(teze)
```

Out[9]:

```
339
```

In [10]:

```
max(teze)
```

Out[10]:

```
80
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [11]:

```
niz = "atacg"
```

In [12]:

```
niz.count('a')
```

Out[12]:

```
2
```

In [ ]:

In [17]:

```
naj_frek = 0
naj_gen = 'ga ni'
for vrstica in open('datoteke/qwerty-dna.txt'):
    ime, sekvenca = vrstica.split()
    trenutna_frek = sekvenca.count('a')
    print(ime, trenutna_frek)
    if trenutna_frek > naj_frek:
        print("  Nov rekord!", trenutna_frek)
        naj_frek = trenutna_frek
        naj_gen = ime
print(f"V genu {naj_gen} se A pojavi {naj_frek}-krat.")
```

```
ASDF13 26
  Nov rekord! 26
SDFG14 31
  Nov rekord! 31
DFGH15 26
FGHJ16 19
GHJK17 32
  Nov rekord! 32
HJKL18 29
ZXCV19 24
XCVB20 21
CVBN21 24
VBNM22 27
V genu GHJK17 se A pojavi 32-krat.
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [19]:

```
seznam = [1,2,3,4]
```

In [27]:

```
seznam[1]
```

```
Out[27]: 2
```

```
In [35]: seznam[10]
```

```
---------------------------------------------------------------------------
-
IndexError                                Traceback (most recent call last
)
<ipython-input-35-f2ea88732e5b> in <module>
----> 1 seznam[10]

IndexError: list index out of range
```

```
In [37]: len(seznam)
```

```
Out[37]: 4
```

```
In [40]: seznam[4]
```

```
---------------------------------------------------------------------------
-
IndexError                                Traceback (most recent call last
)
<ipython-input-40-8a851d9c5d12> in <module>
----> 1 seznam[4]

IndexError: list index out of range
```

```
In [41]: seznam[3]
```

```
Out[41]: 4
```

```
In [ ]:
```

```
In [21]: teze = {'Ana': 72, 'Berta': 85, 'Eva': 50}
```

```
In [22]: teze
```

```
Out[22]: {'Ana': 72, 'Berta': 85, 'Eva': 50}
```

```
In [23]: teze['Ana']
```

```
Out[23]: 72
```

```
In [24]: teze['Eva']
```

```
Out[24]: 50
```

```
In [29]: teze['Martin']
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call last
)
<ipython-input-29-dad95dc2d519> in <module>
----> 1 teze['Martin']

KeyError: 'Martin'
```

In [30]:
```
if 'Martin' in teze:
    print("je noter")
else:
    print("ni ga")
```

```
ni ga
```

In [31]:
```
'Martin' in teze
```

Out[31]:
```
False
```

In [32]:
```
'Ana' in teze
```

Out[32]:
```
True
```

In [43]:
```
teze
```

Out[43]:
```
{'Ana': 72, 'Berta': 85, 'Eva': 50}
```

In [44]:
```
teze['Martin']
```

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call last
)
<ipython-input-44-dad95dc2d519> in <module>
----> 1 teze['Martin']

KeyError: 'Martin'
```

In [45]:
```
teze.get('Martin', 0)
```

Out[45]:
```
0
```

In [46]:
```
teze.get('Martin', 'ga ni')
```

Out[46]:
```
'ga ni'
```

In [47]:
```
stevila = {1: '?', 2: 'prastevilo', 3: 'prastevilo', 4: 'ni prastevilo',
```

In [48]:
```
stevila
```

Out[48]:
```
{1: '?', 2: 'prastevilo', 3: 'prastevilo', 4: 'ni prastevilo', 5: 'prast
evilo'}
```

In [49]:
```
stevila[1]
```

```
Out[49]: '?'

In [50]: stevila[2]

Out[50]: 'prastevilo'

In [51]: teze

Out[51]: {'Ana': 72, 'Berta': 85, 'Eva': 50}

In [52]: teze['Cilka'] = 70

In [53]: teze

Out[53]: {'Ana': 72, 'Berta': 85, 'Eva': 50, 'Cilka': 70}

In [54]: teze['Ana'] = 73

In [62]: teze

Out[62]: {'Berta': 85, 'Eva': 50, 'Cilka': 70}

In [64]: teze = {'Ana': 72, 'Berta': 85, 'Eva': 50}

In [65]: teze

Out[65]: {'Ana': 72, 'Berta': 85, 'Eva': 50}

In [66]: del teze['Ana']

In [67]: teze

Out[67]: {'Berta': 85, 'Eva': 50}

In [68]: del teze['Ana']

         ---------------------------------------------------------------------------
         -
         KeyError                                  Traceback (most recent call last
         )
         <ipython-input-68-61cd90e9374f> in <module>
         ----> 1 del teze['Ana']

         KeyError: 'Ana'

In [69]: teze

Out[69]: {'Berta': 85, 'Eva': 50}

In [70]: teze = {'Ana': 72, 'Berta': 85, 'Eva': 50}

In [ ]:
```

```
In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [71]: teze = {'Ana': 72, 'Berta': 85, 'Eva': 50}

In [72]: teze

Out[72]: {'Ana': 72, 'Berta': 85, 'Eva': 50}

In [73]: teze.keys()

Out[73]: dict_keys(['Ana', 'Berta', 'Eva'])

In [74]: teze.values()

Out[74]: dict_values([72, 85, 50])

In [75]: teze.items()

Out[75]: dict_items([('Ana', 72), ('Berta', 85), ('Eva', 50)])

In [76]: for ime, teza in teze.items():
             print(ime, teza)

         Ana 72
         Berta 85
         Eva 50

In [ ]:

In [ ]:

In [ ]:
```

In [81]:

```
teze_zgodovina = {
    'Ana': [70, 71, 72],
    'Berta': [90, 85],
    'Cilka': [77,  75, 72, 70],
    'Eva': [50, 48, 50]
}
```

In [79]:

```
teze_zgodovina
```

Out[79]:

```
{'Ana': [70, 71, 72],
 'Berta': [90, 85],
 'Cilka': [77, 75, 72, 70],
 'Eva': [50, 48, 50]}
```

In [ ]:

In [91]:

```
for ime, teze in teze_zgodovina.items():
    zacetna_teza = teze[0]
    koncna_teza = teze[-1]
    d = abs(koncna_teza - zacetna_teza)

    if zacetna_teza > koncna_teza:
        print(f"{ime} je shujšala(a) za {d} kg.")
    if zacetna_teza < koncna_teza:
        print(f"{ime} se je zredila(a) za {d} kg.")
    if zacetna_teza == koncna_teza:
        print(f"{ime} se ni spremenil(a). ")
```

```
Ana se je zredila(a) za 2 kg.
Berta je shujšala(a) za 5 kg.
Cilka je shujšala(a) za 7 kg.
Eva se ni spremenil(a).
```

In [93]:

```
for ime, teze in teze_zgodovina.items():
    zacetna_teza = teze[0]
    koncna_teza = teze[-1]
    d = koncna_teza - zacetna_teza

    if zacetna_teza > koncna_teza:
        print(f"{ime} je shujšala(a) za {abs(d)} kg.")
    elif zacetna_teza < koncna_teza:
        print(f"{ime} se je zredila(a) za {abs(d)} kg.")
    else:
        print(f"{ime} se ni spremenil(a). ")
```

```
Ana se je zredila(a) za 2 kg.
Berta je shujšala(a) za 5 kg.
Cilka je shujšala(a) za 7 kg.
Eva se ni spremenil(a).
```

In [ ]:

In [ ]:

In [ ]:

```
In [ ]:

In [ ]:

In [83]: teze = teze_zgodovina['Ana']

In [84]: teze

Out[84]: [70, 71, 72]

In [85]: teze[0]

Out[85]: 70

In [86]: teze[len(teze)-1]

Out[86]: 72

In [87]: teze[-1]

Out[87]: 72

In [88]: teze[-2]

Out[88]: 71

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [95]: imenik = {
             'Marko': ["04127722", "03143123421"],
             'Miha': ["041888292"]
         }

In [96]: imenik
```

```
Out[96]:  {'Marko': ['04127722', '03143123421'], 'Miha': ['041888292']}
```

```
In [98]:  imenik['Miha'] = "nova telefonska"
```

```
In [99]:  imenik
```

```
Out[99]:  {'Marko': ['04127722', '03143123421'], 'Miha': 'nova telefonska'}
```

```
In [100…  imenik = {
              'Marko': ["04127722", "03143123421"],
              'Miha': ["041888292"]
          }
```

```
In [107…  # dodatna stevilka za že obstoječe ime
          imenik['Miha'].append("051888888")
```

```
In [102…  imenik
```

```
Out[102]:  {'Marko': ['04127722', '03143123421'], 'Miha': ['041888292', '051888888'
           ]}
```

```
In [103…  'Blaz' in imenik
```

```
Out[103]:  False
```

```
In [108…  # novo ime in njegova "prva" stevilka
          imenik['Blaz'] = ["01432143124"]
```

```
In [109…  imenik
```

```
Out[109]:  {'Marko': ['04127722', '03143123421'],
            'Miha': ['041888292', '051888888', '051888888'],
            'Blaz': ['01432143124']}
```

```
In [111…  seznam = imenik['Blaz']
```

```
In [112…  seznam
```

```
Out[112]:  ['01432143124']
```

```
In [113…  seznam.append('Nova stevilka')
```

```
In [114…  seznam
```

```
Out[114]:  ['01432143124', 'Nova stevilka']
```

```
In [115…  imenik
```

```
Out[115]:  {'Marko': ['04127722', '03143123421'],
            'Miha': ['041888292', '051888888', '051888888'],
            'Blaz': ['01432143124', 'Nova stevilka']}
```

```
In [ ]:
```

In [ ]:

preberi telefonski imenik in shrani vse telefonske posamezne osebe

In [127…

```
imenik = {}
for vrstica in open('datoteke/telefonske.txt'):
    ime, telefonska = vrstica.split()
    if ime in imenik:
        imenik[ime].append(telefonska)
    else:
        imenik[ime] = [telefonska]
```

In [128…

```
imenik
```

Out[128]:

```
{'Ana': ['0409381326', '0413339231'],
 'Berta': ['0412399483'],
 'Cilka': ['0312791485', '0417721128', '0407721128'],
 'Dani': ['23013905'],
 'Luka': ['0312921789']}
```

In [ ]:

In [143…

```
imenik = {}
for vrstica in open('datoteke/telefonske.txt'):
    ime, telefonska = vrstica.split()
    imenik.setdefault(ime, []).append(telefonska)
```

In [ ]:

In [140…

```
slovar = {'Ana': [1,2,3], 'Berta': [1,2]}
```

In [142…

```
slovar.setdefault('Martin', [])
```

Out[142]:

```
[]
```

In [144…

```
slovar
```

Out[144]:

```
{'Ana': [1, 2, 3], 'Berta': [1, 2], 'Martin': []}
```

In [146…

```
slovar.setdefault('Ana', [100,1001, 10001])
```

Out[146]:

```
[1, 2, 3]
```

In [147…

```
slovar
```

Out[147]:

```
{'Ana': [1, 2, 3], 'Berta': [1, 2], 'Martin': []}
```

In [ ]:

In [ ]:

```
In [ ]:

In [ ]:

In [116…  seznam = [['ena', 'dva', 'tri'], [10,11,12], [0.3, 0.6], ['ena', 2, 'tri'

In [117…  seznam

Out[117]:  [['ena', 'dva', 'tri'], [10, 11, 12], [0.3, 0.6], ['ena', 2, 'tri']]

In [118…  seznam[0].append('stiri')

In [119…  seznam[0]

Out[119]:  ['ena', 'dva', 'tri', 'stiri']

In [120…  seznam

Out[120]:  [['ena', 'dva', 'tri', 'stiri'], [10, 11, 12], [0.3, 0.6], ['ena', 2, 't
           ri']]

In [122…  terke = [('ena', 'dva', 'tri'), (10,11,12), [0.3, 0.6], ['ena', 2, 'tri']

In [123…  terke

Out[123]:  [('ena', 'dva', 'tri'), (10, 11, 12), [0.3, 0.6], ['ena', 2, 'tri']]

In [124…  terke[0].append('stiri')

         ---------------------------------------------------------------------------
         -
         AttributeError                            Traceback (most recent call last
         )
         <ipython-input-124-c2a0f1bab7ab> in <module>
         ----> 1 terke[0].append('stiri')

         AttributeError: 'tuple' object has no attribute 'append'

In [125…  terke[2].append(0.8)

In [126…  terke

Out[126]:  [('ena', 'dva', 'tri'), (10, 11, 12), [0.3, 0.6, 0.8], ['ena', 2, 'tri']
           ]

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```

In [131…

```
# primer rezultata, ki bi radi dobili
frek = {
    'A': 10,
    'T': 7,
    'C': 15,
    'G': 8
}
```

In [159…

```
seq = "ACGAGAGTGCTGCGACGTGCACACAGTG"
```

In [ ]:

```

```

In [160…

```
frek = {}
for znak in seq:
    frek[znak] = frek.get(znak, 0) + 1
```

In [161…

```
frek
```

Out[161]:

```
{'A': 7, 'C': 7, 'G': 10, 'T': 4}
```

In [155…

```
frek.keys()
```

Out[155]:

```
dict_keys(['M', 'a', 'r', 'k', 'o', ' ', 'i', 'n', 'e', 't'])
```

In [ ]:

```

```

In [156…

```
def prestej(niz):
    frek = {}
    for znak in niz:
        frek[znak] = frek.get(znak, 0) + 1
    return frek
```

In [ ]:

```

```

In [162…

```
frek
```

Out[162]:

```
{'A': 7, 'C': 7, 'G': 10, 'T': 4}
```

In [163…

```
prestej('anabanana')
```

Out[163]:

```
{'a': 5, 'n': 3, 'b': 1}
```

In [164…

```
frek
```

Out[164]:

```
{'A': 7, 'C': 7, 'G': 10, 'T': 4}
```

In [ ]:

```

```

In [ ]:

```

```

In [170…

```
a = 4

def mojafun(a): # spremeljive (argumentov in definiranih v funkciji) so l
    a = b
    a = 2
    print('Vrednost a med klicem je', a)

b = "nekaj"
print('Vrednost a pred klicem je', a)
mojafun("eee")
print('Vrednost a po klicu je', a)
```

```
Vrednost a pred klicem je 4
Vrednost a med klicem je 2
Vrednost a po klicu je 4
```

In [ ]:

In [205…

```
niz = "0123456789012"
```

In [206…

```
frek = {}
i = 0
while i < len(niz)-(3-1):
    #print(i, niz[i:i+3])
    kmer = niz[i:i+3]
    frek[kmer] = frek.get(kmer, 0) + 1
    i = i + 3
```

In [207…

```
frek
```

Out[207]:

```
{'012': 1, '345': 1, '678': 1, '901': 1}
```

In [211…

```
# prestej_kmer("ATATATCG", 2), neprekrivajoče
# ->
# {'AT': 3, 'CG': 1}

def prestej_kmer(niz, k):
    frek = {}
    i = 0
    while i < len(niz)-(k-1):
        #print(i, niz[i:i+3])
        kmer = niz[i:i+k]
        frek[kmer] = frek.get(kmer, 0) + 1
        i = i + k
    return frek
```

In [213…

```
prestej_kmer('ATATATCG', 3)
```

Out[213]:

```
{'ATA': 1, 'TAT': 1}
```

In [176…

```
niz[0:3]
```

Out[176]:

```
'012'
```

In [214…

```
# prestej_kmer_prekrivajoce("ATATATCG", 2), neprekrivajoče
# ->
# {'AT': 3, 'TA': 2, 'TC': 1, CG': 1}

def prestej_kmer_prekrivajoce(niz, k):
    frek = {}
    i = 0
    while i < len(niz)-(k-1):
        #print(i, niz[i:i+3])
        kmer = niz[i:i+k]
        frek[kmer] = frek.get(kmer, 0) + 1
        i = i + 1
    return frek
```

In [215…

```
prestej_kmer_prekrivajoce('ATATATCG', 2)
```

Out[215]:

```
{'AT': 3, 'TA': 2, 'TC': 1, 'CG': 1}
```

In [177…

```
niz[3:6]
```

Out[177]:

```
'345'
```

Risanje

In [217…

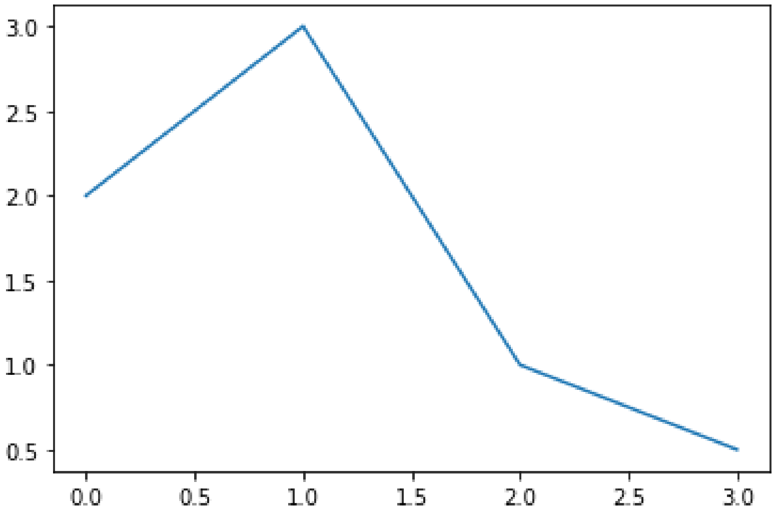
```
import matplotlib.pyplot as plt
```

In [218…

```
plt.plot([2, 3, 1, 0.5])
```

Out[218]:

```
[<matplotlib.lines.Line2D at 0x7f70a7840580>]
```

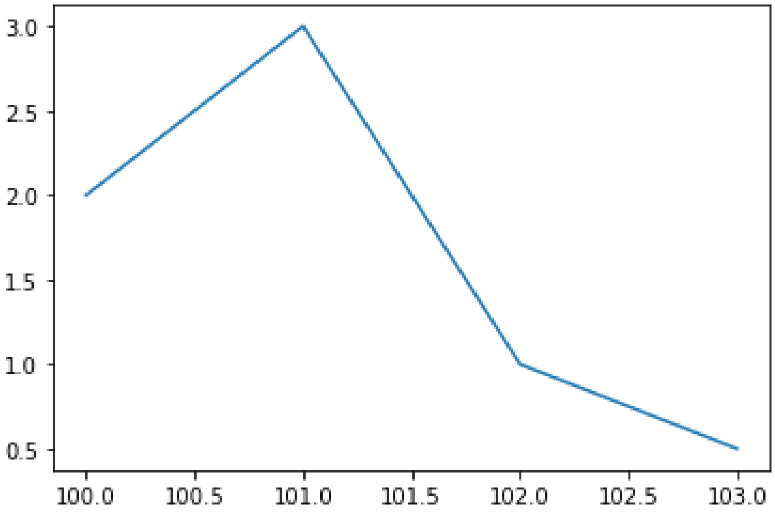


In [219…

```
plt.plot([100, 101, 102, 103], [2, 3, 1, 0.5])
```

Out[219]:

```
[<matplotlib.lines.Line2D at 0x7f70a773ceb0>]
```

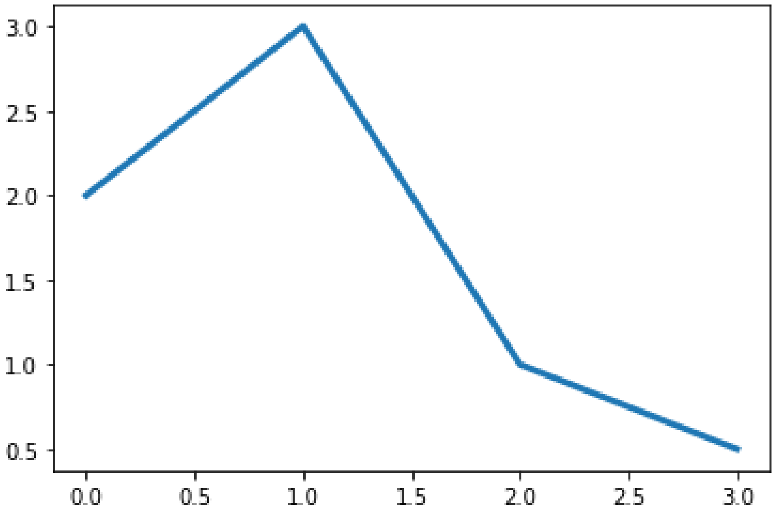


In [220…

```
plt.plot([2, 3, 1, 0.5], lw=3)
```

Out[220]:

```
[<matplotlib.lines.Line2D at 0x7f70a772fb50>]
```

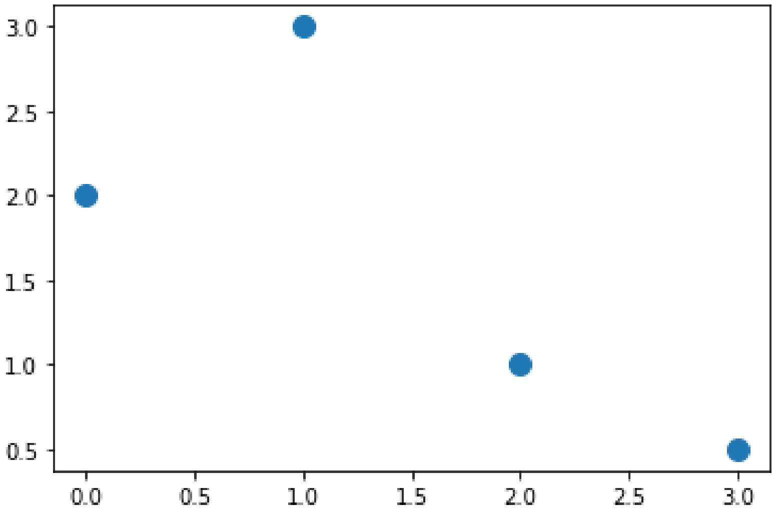


In [223…

```
plt.plot([2, 3, 1, 0.5], 'o', ms=10)
```

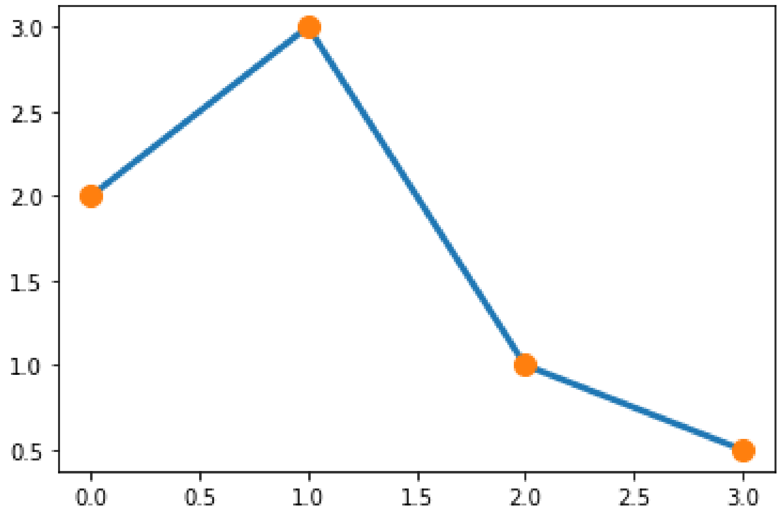
Out[223]:

```
[<matplotlib.lines.Line2D at 0x7f70a75e4d60>]
```

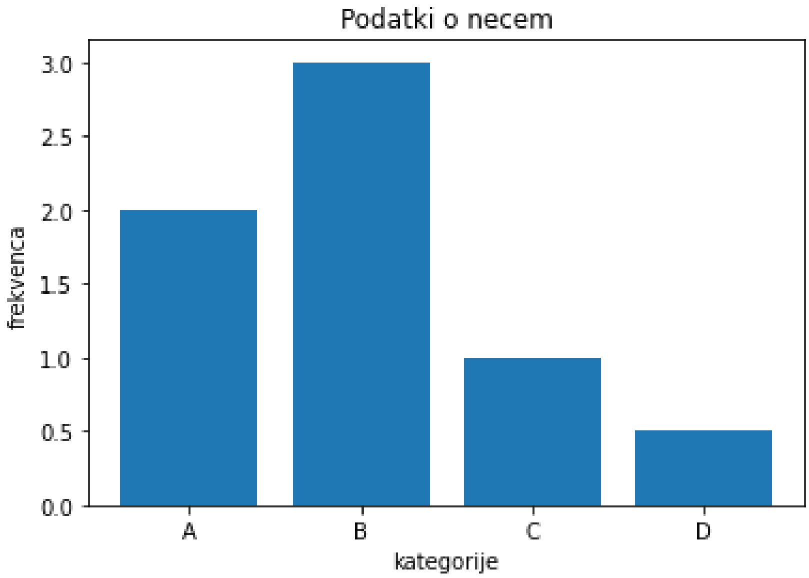


In [226…

```
plt.plot([2, 3, 1, 0.5], lw=3)
plt.plot([2, 3, 1, 0.5], 'o', ms=10)
plt.savefig('mojagraf.png')
```



```
In [231…  plt.bar([0, 1, 2, 3], [2,3,1,0.5])
          plt.title('Podatki o necem')
          plt.xlabel('kategorije')
          plt.ylabel('frekvenca')
          plt.xticks([0, 1, 2, 3], ['A', 'B', 'C', 'D']);
```



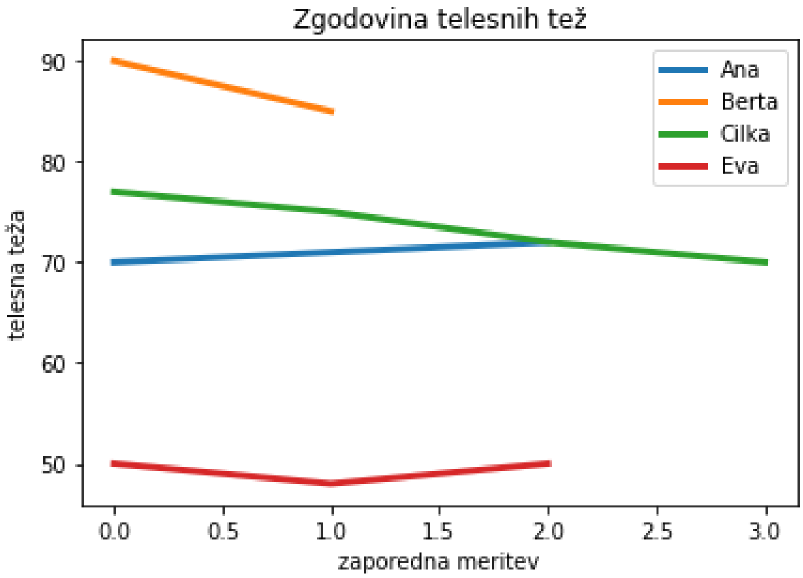
```
In [232…  teze_zgodovina

Out[232]:  {'Ana': [70, 71, 72],
            'Berta': [90, 85],
            'Cilka': [77, 75, 72, 70],
            'Eva': [50, 48, 50]}
```

```
In [236…  for ime, teze in teze_zgodovina.items():
              plt.plot(teze, lw=3, label=ime)

          plt.xlabel('zaporedna meritev')
          plt.ylabel('telesna teža')
          plt.title('Zgodovina telesnih tež')
          plt.legend()

Out[236]:  <matplotlib.legend.Legend at 0x7f70a7186d90>
```


Zgodovina telesnih tež
90
80
70
60
50
telesna teža
0.0
0.5
1.0
1.5
2.0
2.5
3.0
zaporedna meritev
Ana
Berta
Cilka
Eva

In [ ]: