In [1]:
```
import Bio
```

In [2]:
```
from Bio import SeqIO
```

In [7]:
```
for zapis in SeqIO.parse("datoteke/primer.fasta", "fasta"):
    print(zapis.id)
    print(zapis.seq)
    print(len(zapis))
```

```
sek1
ATCGGTGTGCACAGTGTGTCACACAGTGTGCACAGAGAAAGGTGGGTTTATCCACAGAGAGGTGTG
66
sekX
TGTGTGCACAGAGATGTGACAGATGAGATCAACCACACACACAGTGGTGACAGAGACTACATAGTGAGTTGCAC
AGAGAGACATGGGGTG
90
```

In [12]:
```
zapis.name
```

Out[12]:
```
'sekX'
```

In [13]:
```
zapis.id
```

Out[13]:
```
'sekX'
```

In [9]:
```
zapis.description
```

Out[9]:
```
'sekX brezopisa'
```

In [14]:
```
zapis
```

Out[14]:
```
SeqRecord(seq=Seq('TGTGTGCACAGAGATGTGACAGATGAGATCAACCACACACACAGTGGTGACAG
A...GTG', SingleLetterAlphabet()), id='sekX', name='sekX', description='
sekX brezopisa', dbxrefs=[])
```

In [15]:
```
zapis.seq
```

Out[15]:
```
Seq('TGTGTGCACAGAGATGTGACAGATGAGATCAACCACACACACAGTGGTGACAGA...GTG', Sing
leLetterAlphabet())
```

In [16]:
```
str(zapis.seq)
```

Out[16]:
```
'TGTGTGCACAGAGATGTGACAGATGAGATCAACCACACACACAGTGGTGACAGAGACTACATAGTGAGTTG
CACAGAGAGACATGGGGTG'
```

In [17]:
```
zapis
```

Out[17]:
```
SeqRecord(seq=Seq('TGTGTGCACAGAGATGTGACAGATGAGATCAACCACACACACAGTGGTGACAG
A...GTG', SingleLetterAlphabet()), id='sekX', name='sekX', description='
sekX brezopisa', dbxrefs=[])
```

In [18]:
```
zapis[2:8]
```

Out[18]:
```
SeqRecord(seq=Seq('TGTGCA', SingleLetterAlphabet()), id='sekX', name='se
kX', description='sekX brezopisa', dbxrefs=[])
```

In [20]:
```
zapisi = []
for zapis in SeqIO.parse("datoteke/primer2.fasta", "fasta"):
    zapisi.append(zapis[:5])

SeqIO.write(zapisi, 'zacetki.fasta', 'fasta')
```

Out[20]:
```
3
```

In [21]:
```
SeqIO.write([zapis[:5] for zapis in SeqIO.parse("datoteke/primer2.fasta",
```

Out[21]:
```
3
```

Za vsak zapis izpiši ID zapisa in frekvenco nukleotidov v sekvenci.

In [22]:
```
for zapis in SeqIO.parse("datoteke/primer2.fasta", "fasta"):

```

```
  File "<ipython-input-22-204afea04e63>", line 2

    ^
SyntaxError: unexpected EOF while parsing
```

In [23]:
```
niz = "ATGTGTACAGTGACGACACA"
```

In [25]:
```
niz.count("T")
```

Out[25]:
```
4
```

In [30]:
```
def prestej(niz):
    frekvenca = {}
    for znak in niz:
        if znak not in frekvenca:
            frekvenca[znak] = 0
        frekvenca[znak] = frekvenca[znak] + 1
    return frekvenca
```

In [31]:
```
prestej(niz)
```

Out[31]:
```
{'A': 7, 'T': 4, 'G': 5, 'C': 4}
```

In [35]:
```
for zapis in SeqIO.parse("datoteke/primer2.fasta", "fasta"):
    print(zapis.id)
    print(prestej(str(zapis.seq)))
```

```
sek1
{'A': 17, 'T': 15, 'C': 11, 'G': 23}
sekX
{'T': 16, 'G': 28, 'C': 16, 'A': 30}
```

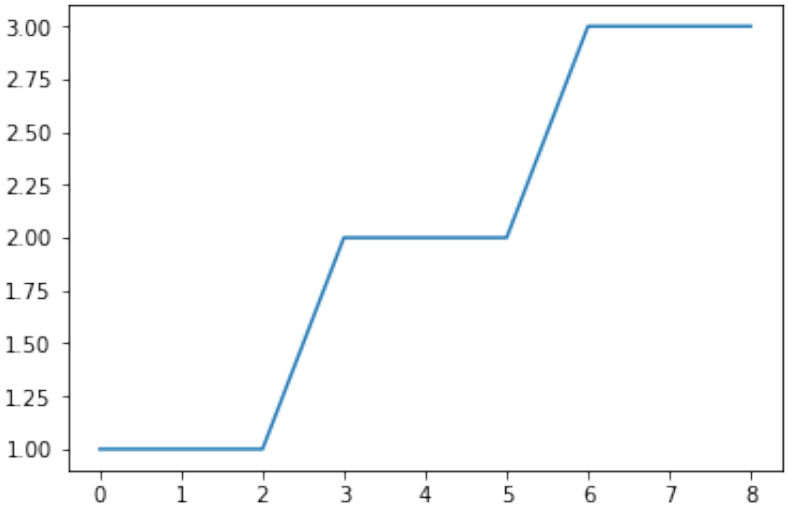
In [32]:
```
str(zapis.seq)
```

Out[32]:
```
'ATGTGAGAGTGTCCC'
```

In [36]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [37]:
```
plt.plot([1, 1, 1, 2, 2, 2, 3, 3, 3])
```

Out[37]:
```
[<matplotlib.lines.Line2D at 0x7fa126ac0860>]
```



In [59]:
```
import random
niz = "".join([random.choice(["A", "T", "C", "G"]) for x in range(500)])
```

In [60]:
```
niz
```

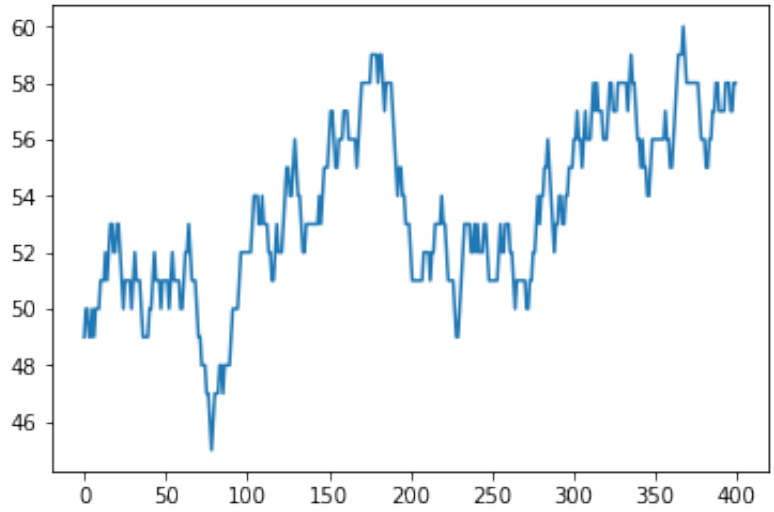
Out[60]:
```
'TACAAGTATTAGAGTATGGTACCCAGCTCTTGGCCGGCAAATACCTGACAACTTCACTCTTTGAGGCTGCG
GACGTCGTATATGCTGCGTAGTCATCTGCCTATCAGATCACGTGGATGCTATTCGCAACGTCGTAGGACTCC
TGTTCGAAACGAACTAACGGCAAGTTTGAAGTAATGGAAGCTCGCGCCCACCGCTCGCAGGGAAACTTCATT
CGGAGACGCCACCCAATATTCCCTCTGCCAGCCGCGTAATGGTGACTGGAAACCGCTACTGAAGAGAATCCG
CTAATCGTGAAGAAAGCGTTCCTTTCTCCCCGAATTGGTTTCGGCGATCTGCTCTCGCCAACCGTACCTCCG
AATTTGTAGCGATGCCGCGTCGGTCTAATGTGAAGGCCAGGAGTCAGGACAGGTCAGCACTGGTAGTCAGGG
CTCCTAAATTGACAGCGCCCTACCCACAAGCCCCTGTTGTACGCCTTACAGGCGAGATTCACCGTCCGA'
```

In [61]:
```
def gc(sek):
    return sek.count("G") + sek.count("C")

vrednosti = []
for x in range(len(niz)-100):
    v = gc(niz[x:x+100])
    vrednosti.append(v)
```

In [62]:
```
plt.plot(vrednosti)
```

Out[62]:
```
[<matplotlib.lines.Line2D at 0x7fa1268176d8>]
```

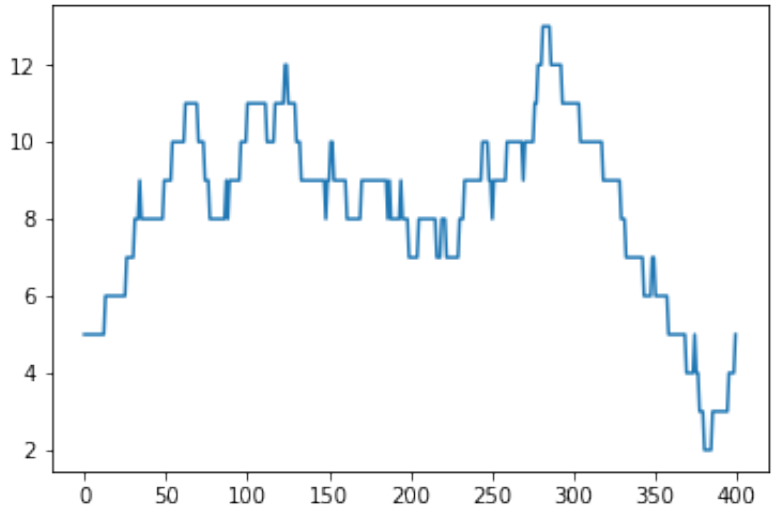


In [63]:
```
def CpG(sek):
    return sek.count("CG")

vrednosti = []
for x in range(len(niz)-100):
    v = CpG(niz[x:x+100])
    vrednosti.append(v)
```

In [64]:
```
plt.plot(vrednosti)
```

Out[64]:
```
[<matplotlib.lines.Line2D at 0x7fa1267fed68>]
```



In [80]:
```
import urllib
import gzip

f = gzip.open(urllib.request.urlopen('ftp://ftp.sra.ebi.ac.uk/vol1/fastq/

for rec in list(SeqIO.parse(f, 'fastq'))[:3]:
    print(rec.id)
    print(repr(rec.seq))
    print(rec.letter_annotations["phred_quality"])
    print(len(rec))
```

```
SRR020192.1
Seq('GATGACGGTGTCTACATTGTTCCCGACCACTCATCTCCTCTGTCATGCCCGAAA...CGT', Single
LetterAlphabet())
[24, 23, 27, 30, 30, 30, 23, 23, 24, 23, 23, 30, 28, 27, 25, 25, 27, 27, 2
7, 22, 22, 24, 18, 18, 18, 30, 19, 19, 23, 23, 30, 30, 32, 32, 32, 30, 24,
23, 23, 27, 30, 32, 30, 32, 29, 28, 28, 17, 17, 17, 17, 24, 17, 17, 13, 15
, 17, 25, 25, 24, 24, 23, 27, 27, 15, 15, 15, 15, 15, 17, 17, 11, 15, 15]
74
SRR020192.2
Seq('GATGACGGTGTCTACATCGTTCCACCACTCATCTCCTCTGTCATGCCCGAAAGT...CCC', Single
LetterAlphabet())
[27, 27, 27, 30, 30, 30, 23, 23, 24, 27, 27, 30, 28, 27, 27, 27, 27, 30, 3
0, 27, 27, 27, 27, 30, 23, 23, 23, 23, 30, 30, 32, 32, 30, 30, 27, 27, 27,
27, 30, 29, 28, 29, 29, 29, 29, 17, 15, 15, 15, 15, 15, 15, 15, 17, 26, 15
, 15, 15, 15, 27, 27, 15, 15, 15, 15, 15]
66
SRR020192.3
Seq('GACGACGGTGTCTACATCGTTCCACCACTCATCTCCTCTGTCATGCCCAAAGTC...CGT', Single
LetterAlphabet())
[32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 4
0, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40,
37, 37, 37, 32, 37, 37, 34, 34, 30, 30, 30, 30, 30, 30, 31, 32, 33, 30, 30
, 30, 28, 30, 29, 24, 24, 18, 18, 19, 17, 22, 24, 24, 21, 21, 23, 27, 23,
24, 22, 22, 15, 23, 23, 21, 21, 24, 24, 25, 15, 15, 15, 11, 11, 15, 15, 23
, 24, 24, 23, 11, 11, 15, 15, 15, 15, 15, 15, 21, 15, 11, 11, 13, 13, 13]
111
```

In [69]:
```
len(rec.seq)
```

Out[69]:
```
111
```

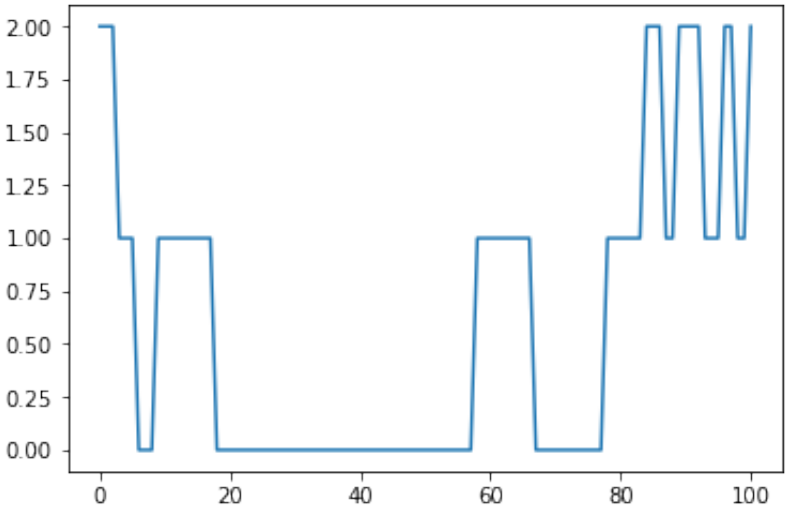
In [70]:
```
niz = str(rec.seq)
```

In [73]:
```
def CpG(sek):
    return sek.count("CG")

vrednosti = []
for x in range(len(niz)-10):
    v = CpG(niz[x:x+10])
    vrednosti.append(v)
```

In [74]:
```
plt.plot(vrednosti)
```

Out[74]:
```
[<matplotlib.lines.Line2D at 0x7fa12334d5f8>]
```



In [82]:
```
rec
```

Out[82]:
```
SeqRecord(seq=Seq('GACGACGGTGTCTACATCGTTCCACCACTCATCTCCTCTGTCATGCCCAAAGT
C...CGT', SingleLetterAlphabet()), id='SRR020192.3', name='SRR020192.3',
description='SRR020192.3 E0LM4JH01BQOOG/2', dbxrefs=[])
```

In [85]:
```
quals = rec.letter_annotations["phred_quality"]
```

In [90]:
```
quals[-10:]
```

Out[90]:
```
[15, 15, 15, 21, 15, 11, 11, 13, 13, 13]
```

In [89]:
```
str(rec.seq)[-10:]
```

Out[89]:
```
'AAACGAACGT'
```

In [91]:
```
niz = "tomaz"
```

In [92]:
```
niz[:-1]
```

Out[92]:
```
'toma'
```

In [93]:
```
vrednosti = [1,10,16,7,5]
```

In [94]:
```
vrednosti[-1]
```

Out[94]:
```
5
```

In [95]:
```
while quals[-1] < 20:
    quals = quals[:-1]
    rec = rec[:-1]
```

In [96]:
```
quals[-5:]
```

Out[96]:
```
[15, 15, 15, 15, 21]
```

In [97]:
```
str(rec.seq)[-10:]
```

Out[97]: 'ACCGACAAAC'

In [ ]:
```
def trim(rec, qual_th=20):
    quals = rec.letter_annotations["phred_quality"]
    while quals and quals[-1] < 20:
        quals = quals[:-1]
        rec = rec[:-1]
    return rec
```

In [98]:
```
seznam = []
```

In [99]:
```
seznam[-1]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-99-b9afe8a902f1> in <module>()
----> 1 seznam[-1]

IndexError: list index out of range
```

In [100…
```
seznam[0]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-100-9cf8dcef6c15> in <module>()
----> 1 seznam[0]

IndexError: list index out of range
```

In [103…
```
seznam = [1,2,2]
```

In [104…
```
if seznam: print("seznam ni prazen")
```

```
seznam ni prazen
```

Dostop do NCBI

In [105…
```
from Bio import SeqIO
```

In [107…
```
from Bio import Entrez

Entrez.email = 'moj.mail@nanslovu.si'
```

In [108…
```
handle = Entrez.efetch(db='nucleotide', rettype='gb', id='NC_012920.1')
```

In [109…
```
rec = SeqIO.read(handle, "gb")
```

In [110…
```
len(rec.seq)
```

Out[110]: 16569

In [111…
```
rec.id
```

Out[111]: 'NC_012920.1'

In [112…
```
rec.description
```

Out[112]: 'Homo sapiens mitochondrion, complete genome'

In [113…
```
rec.seq
```

Out[113]: Seq('GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCATTTGG...ATG', IUPACAmbiguousDNA())

In [115…
```
len(rec.features)
```

Out[115]: 105

In [116…
```
rec.features[0]
```

Out[116]: SeqFeature(FeatureLocation(ExactPosition(0), ExactPosition(16569), strand=1), type='source')

In [117…
```
rec.features[1]
```

Out[117]: SeqFeature(CompoundLocation([FeatureLocation(ExactPosition(0), ExactPosition(576), strand=-1), FeatureLocation(ExactPosition(16023), ExactPosition(16569), strand=-1)], 'join'), type='D-loop', location_operator='join')

In [118…
```
print(rec.features[1])
```

```
type: D-loop
location: join{[0:576](-), [16023:16569](-)}
qualifiers:
```

In [119…
```
rec.features[1].type
```

Out[119]: 'D-loop'

In [120…
```
rec.features[1].location
```

Out[120]: CompoundLocation([FeatureLocation(ExactPosition(0), ExactPosition(576), strand=-1), FeatureLocation(ExactPosition(16023), ExactPosition(16569), strand=-1)], 'join')

In [121…
```
rec.seq
```

Out[121]: Seq('GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCATTTGG...ATG', IUPACAmbiguousDNA())

In [123…
```
dloop = rec.features[1].extract(rec)
```

In [124…]
```
dloop
```

Out[124]:
```
SeqRecord(seq=Seq('TGTGGGGGGTGTCTTTGGGGTTTGGTTGGTTCGGGGTATGGGGTTAGCAGCGG
T...GAA', IUPACAmbiguousDNA()), id='<unknown id>', name='<unknown name>'
, description='<unknown description>', dbxrefs=[])
```

In [125…]
```
len(dloop)
```

Out[125]:
```
1122
```

In [126…]
```
dloop.seq
```

Out[126]:
```
Seq('TGTGGGGGGTGTCTTTGGGGTTTGGTTGGTTCGGGGTATGGGGTTAGCAGCGGT...GAA', IUPA
CAmbiguousDNA())
```

In [127…]
```
rec
```

Out[127]:
```
SeqRecord(seq=Seq('GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCATTTG
G...ATG', IUPACAmbiguousDNA()), id='NC_012920.1', name='NC_012920', desc
ription='Homo sapiens mitochondrion, complete genome', dbxrefs=['BioProj
ect:PRJNA30353'])
```

In [140…]
```
codons = {}
for feature in rec.features:
    if feature.type == 'CDS':
        sekvenca = str(feature.extract(rec).seq)
        for x in range(0, len(sekvenca)-2, 3):
            cod = sekvenca[x: x+3]
            if cod not in codons:
                codons[cod] = 0
            codons[cod] += 1
```

In [141…]
```
codons
```

Out[141]:
```
{'ATA': 167,
 'CCC': 119,
 'ATG': 40,
 'GCC': 124,
 'AAC': 132,
 'CTC': 167,
 'CTA': 276,
 'ATT': 124,
 'GTA': 70,
 'ATC': 196,
 'GCA': 80,
 'TTC': 139,
 'CTT': 65,
 'ACC': 155,
 'GAA': 64,
 'CGA': 28,
 'AAA': 85,
 'GGC': 87,
 'TAT': 46,
 'CAA': 82,
 'CGC': 26,
 'GTT': 31,
```

```
'TAC': 89,
'GGG': 34,
'GCT': 43,
'GAC': 51,
'GAG': 24,
'ACA': 134,
'TCT': 32,
'CCG': 7,
'TTA': 73,
'TGA': 93,
'CTG': 45,
'GTC': 48,
'TTT': 77,
'AGC': 39,
'TCA': 83,
'AGT': 14,
'TCC': 99,
'CAC': 79,
'CCA': 52,
'TTG': 18,
'ACT': 52,
'GGA': 67,
'CCT': 41,
'AAG': 10,
'AAT': 32,
'GCG': 8,
'TCG': 7,
'CGT': 7,
'ACG': 10,
'GGT': 24,
'CAG': 8,
'TGG': 11,
'CAT': 18,
'GAT': 15,
'GTG': 18,
'TGC': 17,
'AGA': 1,
'TGT': 5,
'TAG': 2,
'TAA': 3,
'CGG': 2,
'AGG': 1}
```

In [142…
```
zap1 = 'abbaac'
zap2 = 'aabcdc'
```

In [146…
```
list(zip(zap1, zap2))
```

Out[146]:
```
[('a', 'a'), ('b', 'a'), ('b', 'b'), ('a', 'c'), ('a', 'd'), ('c', 'c')]
```

In [148…
```
enakih = 0
for v1, v2 in zip(zap1, zap2):
#    if v1 == v2:
#        enakih += 1
    enakih += v1 == v2
```

In [149…
```
enakih
```

Out[149]: 3

In [150…
```
[v1 == v2 for v1, v2 in zip(zap1, zap2)]
```

Out[150]: [True, False, True, False, False, True]

In [151…
```
zap1
```

Out[151]: 'abbaac'

In [152…
```
zap2
```

Out[152]: 'aabcdc'

In [153…
```
sum([v1 == v2 for v1, v2 in zip(zap1, zap2)])
```

Out[153]: 3

In [154…
```
sum([1,0,1,0,0,1])
```

Out[154]: 3

In [ ]:
```

```