S10

March 21, 2023

1 38C

1.1 1.

[116]:

```
def kennitala(string):

    string = list(map(int,' '.join(str(string)).split()))
    calc = [3,2,7,6,5,4,3,2]

    sum = 0

    for i in range(8):
        sum += string[i]*calc[i]

    randomnumb = sum%11
    if randomnumb==0:
        return string[8] == 0
    if randomnumb == 1:
        return False
    else:
        randomnumb1 = 11-randomnumb
        return randomnumb1 == string[8]

print(kennitala("2411932769"))
print(kennitala("704012830"))
```

```
True
False
```

1.2 2.

[119]:

```
import numpy as np

A = np.loadtxt("https://cs.hi.is/python/kennitolur.txt", delimiter=',',␣
 ↪skiprows=1, dtype=str).T

Name = A[0].tolist()
ExamNumber = A[1].tolist()
```

```
IdNumber = A[2].tolist()

print("Name                             Exam number  Identity number")
for i in range(len(Name)):
    print(str(Name[i]).ljust(34), str(ExamNumber[i]).ljust(12), IdNumber[i])
```

```
Name                             Exam number  Identity number
Sigrún Jónsdóttir                  0176         2903993279
Kristín Fjóludóttir                0542         1206972699
Birta Lárusdóttir                  0970         2605973109
Erla Ýr Guðnadóttir                1419         1210012330
Anton Ingi Þórsson                 4854         2308984059
Íris María Birgisdóttir            5469         1603903879
Ívar Sigurðsson                    6324         1309932659
Ágúst Guðni Ingason                6558         1601013180
Steinunn Guðlaug Gunnarsdóttir     7923         2006002580
Eydís Þorsteinsdóttir              8003         1703012420
Signý Guðrún Pálsdóttir            8148         2012012410
Elías Ari Heimisson                9058         0704012830
Mark Johnson                       9576         0403983099
Víðir Kristjánsson                 9595         2304003180
Jónas Valdimarsson                 9706         0706012300
```

1.3 3.

[159]:

```
for j,i in enumerate(IdNumber):
    if kennitala(i) == False:
        print("The identity number", i, "is incorrect and is associated with␣
 ↪the name", Name[j])
```

```
The identity number 0704012830 is incorrect and is associated with the name
Elías Ari Heimisson
```

1.4 4.

[121]:

```
import datetime
def checkdate(day,month,year):
    correctDate = None
    try:
        newDate = datetime.datetime(year, month, day)
        correctDate = True
    except ValueError:
        correctDate = False
    return correctDate



months = ['janúar', 'febrúar', 'mars', 'apríl', 'maí', 'júní', 'júlí', 'ágúst',␣
 ↪'september', 'október', 'nóvember', 'desember']
```

```
def checkidentity(checklist):
    for k in checklist:
        day = int(str(k)[:2])
        month = int(str(k)[2:4])
        year = int(str(k)[4:6])
        if checkdate(day, month, year):
            if year<23:
                year = str(k)[4:6]
                print(f"{day}. {months[month-1]} 20{year}")
            else:
                year = str(k)[4:6]
                print(f"{day}. {months[month-1]} 19{year}")
        else:
            print("This date of birth is not valid")

checkidentity(IdNumber)
print("My identity number is 241193-2769.", end=" ")
checkidentity([2411932769])
```

```
29. mars 1999
12. júní 1997
26. maí 1997
12. október 2001
23. ágúst 1998
16. mars 1990
13. september 1993
16. janúar 2001
This date of birth is not valid
17. mars 2001
20. desember 2001
7. apríl 2001
4. mars 1998
This date of birth is not valid
7. júní 2001
My identity number is 241193-2769. 24. nóvember 1993
```

2 D.

2.1 1.

[126]:
```
NameExam = dict(zip(Name, ExamNumber))
```

2.2 2.

[37]:
```
def íslenska(s):
   # notað sem 'key' í sort eða sorted til að raða í íslenska stafrófsröð,
   # t.d. print(sorted(['ár','bára','bali','akur'], key=íslenska))"""
   return [íslenska.k.get(c.lower(),0) for c in s]
íslenska.a = list('aábcdðeéfghiíjklmnoópqrstuúvwxyýzþæö')
íslenska.k = dict(zip(íslenska.a, range(1,len(íslenska.a)+1)))
IcelandicNameExam = sorted(NameExam, key=íslenska)
SortedDict = {i: NameExam[i] for i in IcelandicNameExam}
print("Name                              Exam number")
for i in SortedDict:
   print(i.ljust(34), SortedDict[i])
```

```
Name                              Exam number
Anton Ingi Þórsson                 4854
Ágúst Guðni Ingason                6558
Birta Lárusdóttir                  970
Elías Ari Heimisson                9058
Erla Ýr Guðnadóttir                1419
Eydís Þorsteinsdóttir              8003
Íris María Birgisdóttir            5469
Ívar Sigurðsson                    6324
Jónas Valdimarsson                 9706
Kristín Fjóludóttir                542
Mark Johnson                       9576
Signý Guðrún Pálsdóttir            8148
Sigrún Jónsdóttir                  176
Steinunn Guðlaug Gunnarsdóttir     7923
Víðir Kristjánsson                 9595
```

2.3 3.

[42]:
```
P = [2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97]

nrp = {}
for i in range(len(P)):
    nrp[P[i]] = i+1

def prime_lookup(num, nrp):
    if num > 99:
        return f"{num} er of stór"
    elif num in nrp:
        return f"{num} er {nrp[num]}. prímtalan"
    else:
        return f"{num} er ekki prímtala"

print(prime_lookup(13, nrp))
print(prime_lookup(16, nrp))
print(prime_lookup(103, nrp))
```

```
13 er 6. prímtalan
16 er ekki prímtala
103 er of stór
```

3 VV3. Fall Rosenbrocks

3.1 1.

[206]:

```
def rosen(x,y):
    return (1-x)**2+100*(y-x**2)**2

print(round(rosen(-1.2,1),1))
```

```
24.2
```

3.2 2.

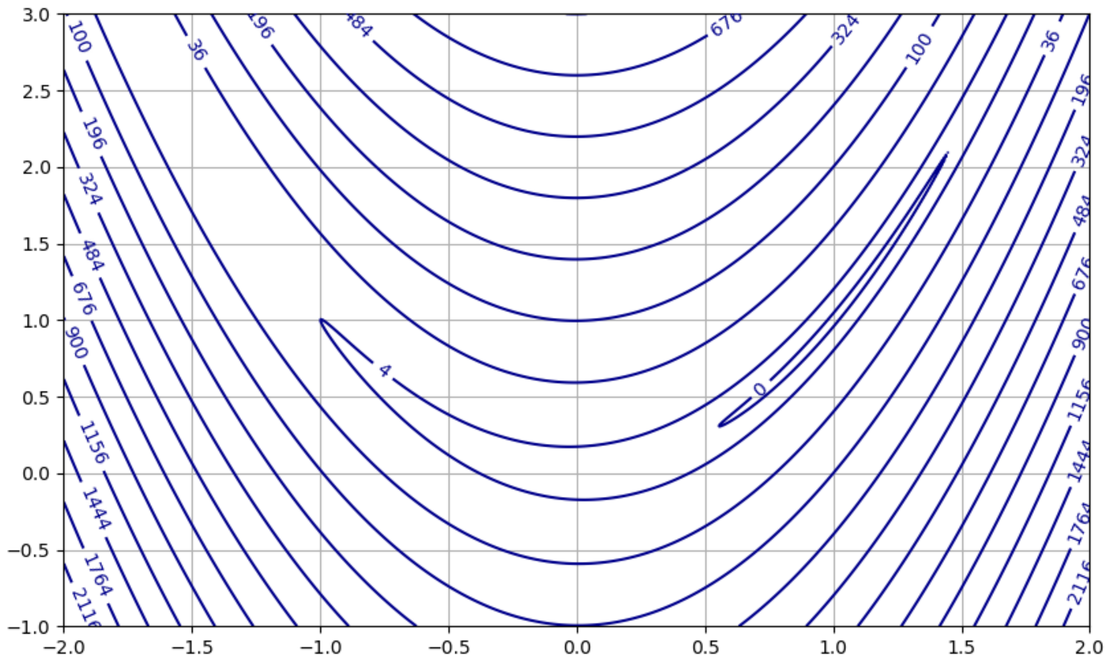
[199]:

```
import matplotlib.pyplot as plt



plt.figure(figsize=(10,6))
xlist = np.linspace(-2.0, 2.0, 500)
ylist = np.linspace(-1.0, 3.0, 500)
z = np.array([[rosen(xi,yj) for xi in xlist] for yj in ylist])
levels = np.append([0.2], np.arange(2,50,4)**2)
c =plt.contour(xlist,ylist, z, levels=levels, colors='darkblue')

plt.clabel(c, fmt="%.0f")
plt.grid(True)
plt.show()
```



4 VV6. Hlutapróf 2 vorið 2021

4.1 1.

```
[169]: def derivative(x, y):
           df_dx = 2*x*y + 2*y**2 - 3*y
           df_dy = x**2 + 4*x*y - 3*x
           return (df_dx, df_dy)

       derivative(1,1)
```

```
[169]: (1, 2)
```

4.2 2.

```
[202]: print(np.dot(np.array(derivative(1,1)), np.array([-1,-1])))
       print("Ég er fæddur 24. Nóv 1993 og ætla að nota 2,4 og 9,3.")
       print(np.dot(np.array(derivative(2,4)), np.array([9,3])))
```

```
-3
Ég er fæddur 24. Nóv 1993 og ætla að nota 2,4 og 9,3.
414
```

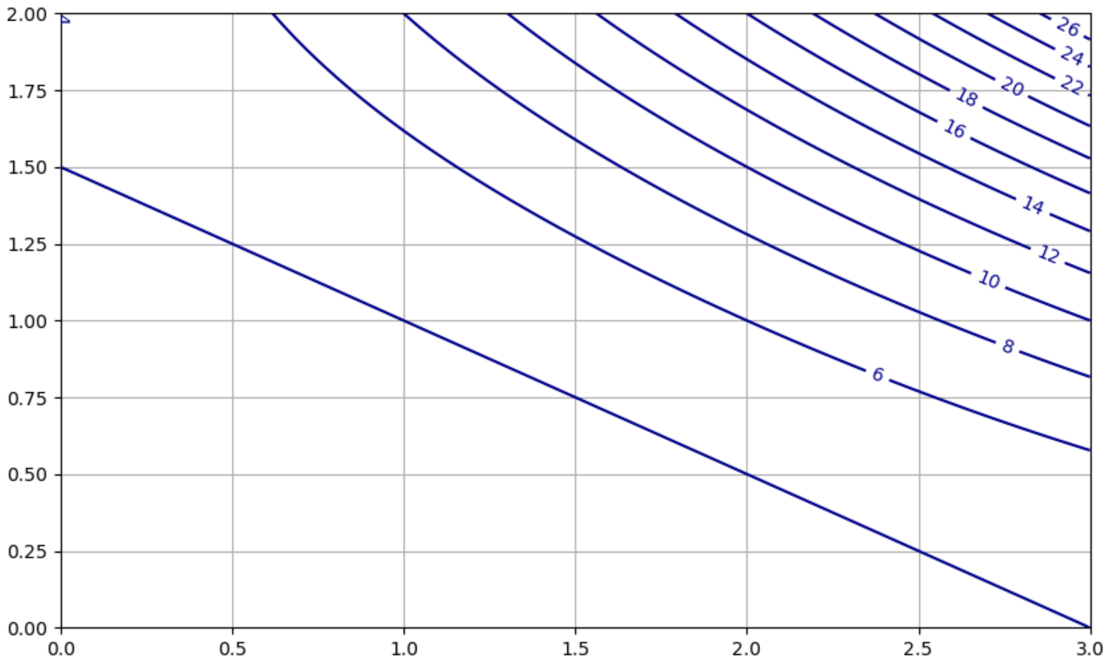
4.3 3.

[205]:

```
def calc(x,y):
    return x**2*y+2*x*y**2-3*x*y+4

plt.figure(figsize=(10,6))
xlist = np.linspace(0, 3.0, 500)
ylist = np.linspace(0, 2.0, 500)
z = np.array([[calc(xi,yj) for xi in xlist] for yj in ylist])
# levels = np.append([0.2], np.arange(2,50,4)**2)
c =plt.contour(xlist,ylist, z, 12, colors='darkblue')

plt.clabel(c, fmt="%.0f")
plt.grid(True)
plt.show()
```



2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.0
0.5
1.0
1.5
2.0
2.5
3.0
4
6
8
10
12
14
16
18
20
22
24
26