Viikko 35 -tehtävät

Tehtävä 1

```
print('Hello world')
```

Tehtävä 2

```
a = int(input("a: "))
b = int(input("b: "))

if a > b:
    print(f'{a} on isompi')
elif b > a:
    print(f'{b} on isompi')
else:
    print('yhtäsuuret')
```

Tehtävä 3

```
import random

a = random.randint(0, 100)
print(a)
b = random.randint(0, 100)
print(b)

if (a > b):
    print(f'{a} on isompi')

elif (b > a):
    print(f'{b} on isompi')

else:
    print('yhtäsuuret')
```

Tehtävä 4

```
def sumAndPrint(num1, num2):
    sum = num1 + num2
    print(f'Lukujen {num1} ja {num2} summa on: {sum}')

sumAndPrint(9, 1)
```

Tehtävä 5

```
import random

userAnswers = []
correctAnswers = []
questionStrings = []

NUM_QUESTIONS = 10

for i in range(NUM_QUESTIONS):
    a = random.randint(0, 10)
    b = random.randint(0, 10)

    userAnswer = int(input(f'{a} * {b} = '))

    questionStrings.append(f'{a} * {b}')
    userAnswers.append(userAnswer)
    correctAnswers.append(a * b)

correctCount = 0
for i in range(NUM_QUESTIONS):
    if userAnswers[i] == correctAnswers[i]:
        print(f'Oikein :-) {questionStrings[i]} =
{correctAnswers[i]}')
        correctCount += 1
    else:
        print(f'Väärin :-( Oikea vastaus on: {questionStrings[i]} =
{correctAnswers[i]}')

print(f'Sait {correctCount}/{NUM_QUESTIONS} oikein!')
```

Tehtävä 6

```
class Murtoluku:
    def __init__(self, os, nim):
        self.os = os
        self.nim = nim

    def tulosta(self):
        print(f'{self.os} / {self.nim}')

    def gcd(self,a, b):
        if b == 0:
            return a
        else:
            return self.gcd(b, a % b)

    def sievenna(self):
        gcd = self.gcd(self.os, self.nim)
        self.nim //= gcd
        self.os //= gcd

ml = Murtoluku(34562, 311058)

ml.tulosta()
ml.sievenna()
ml.tulosta()
```

Tehtävä 7

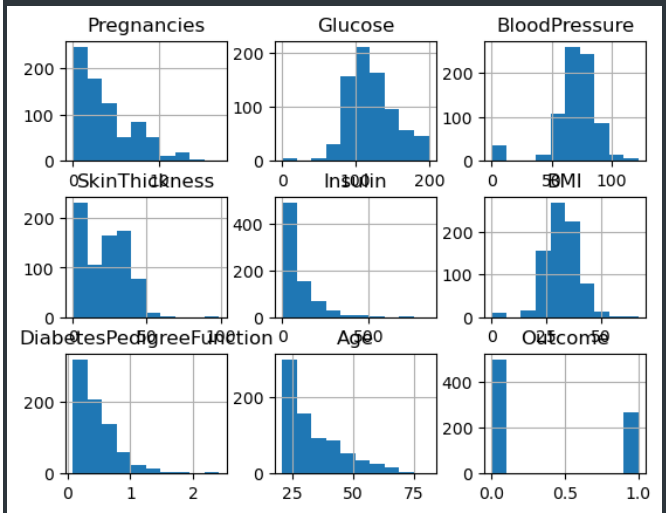
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb

df = pd.read_csv('./work/viikko1/datasets/diabetes.csv')

desc = df.describe()
plt.show()
```

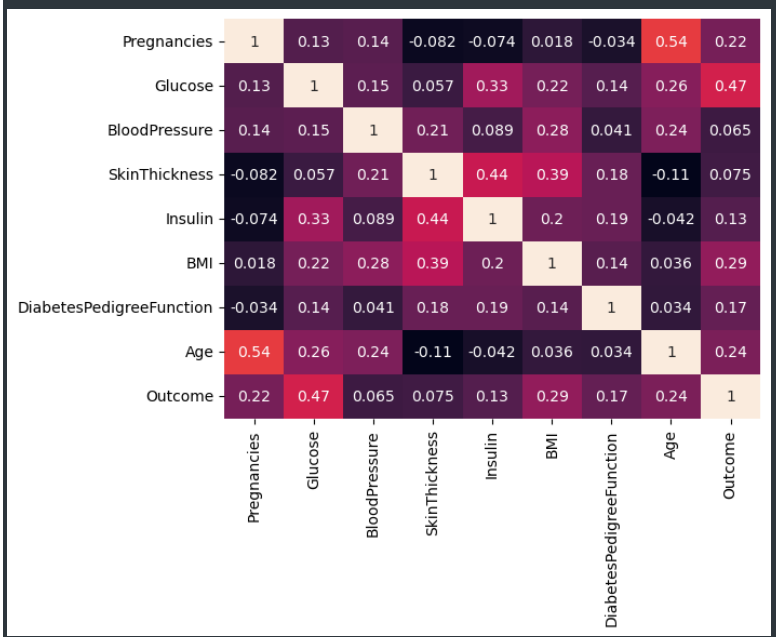
	index	Pregnan...	*Glucose*	Blood...	SkinT...	Insulin	BMI	Diabe...	Age	Outco...
0	count	768	768	767	768	768	768	768	768	768
7	max	17	199	122	99	846	67.1	2.42	81	1
6	75%	6	140.25	80	32	127.25	36.6	0.62625	41	1
1	mean	3.84505208...	120.8945...	69.10169...	20.53645...	79.79947...	31.99257...	0.471876...	33.24088...	0.348958...
5	50%	3	117	72	23	30.5	32	0.3725	29	0
4	25%	1	99	62	0	0	27.3	0.24375	24	0
2	std	3.36957806...	31.97261...	19.36815...	15.95221...	115.2440...	7.884160...	0.331328...	11.76023...	0.476951...
3	min	0	0	0	0	0	0	0.078	21	0

```
df.hist()
plt.show()
```



Tehtävä 8

```
corr = df.corr()
sb.heatmap(corr, annot=True, cbar=False)
plt.show()
```



Tehtävä 9

```
ageGroup = df['Age'].value_counts()
```

	Age	count
0	22	72
1	21	63
2	25	48
3	24	46
4	23	38
5	28	35
6	26	33
7	27	32
8	29	29
9	31	24
10	41	22
11	30	21

```
diabetesCount = df['Outcome'].value_counts()
```

	Outcome	count
0	0	500
1	1	268

Tehtävä 10

```
nullCount = df.isna().sum()
```

	index	0
0	Pregnancies	0
1	Glucose	0
2	BloodPressure	1
3	SkinThickness	0
4	Insulin	0
5	BMI	0
6	DiabetesPedigreeFunction	0
7	Age	0
8	Outcome	0