

Övningsprov

Uppgift 6-7 är VG-uppgifter. För G krävs 50% av totala poängen och för VG krävs 70% av totala poängen varav 50% av VG-poängen.

Uppgift 1 (3p)

Förklara vad den här koden gör och vad som ritas ut:

```
1. import matplotlib.pyplot as plt
2.
3. x = list(range(10))
4. y = [i**2 for i in x]
5. fig, ax = plt.figure(), plt.axes()
6. ax.plot(x,y)
7. ax.set(title="Plot", xlabel="x", ylabel="y")
```

Skriv en kort kommentar för varje rad.

Uppgift 2 (3p)

Simulera 1000000 tärningskast och beräkna antalet ettor, tvåor, treor, fyror, femmor och sexor. Spara resultatet i en lämplig datastruktur.

Uppgift 3 (3p)

Skriv en funktion som tar in en radie som inparameter och som returnerar volymen av en sfär. Volymen för en sfär är:

$$V = \frac{4\pi r^3}{3}$$

Låt användaren mata in ett tal och kör funktionen. Tänk på att ta med lämplig felhantering.

Uppgift 4 (3p)

Läs in en textfil som heter bamba.txt och innehåller:

```
v42
Chili sin carne
Pannkakor
Ärtsoppa
Kebab
Vegetarisk lasagne
```

Läs in texterna och skriv ut dem i terminalen.

Uppgift 5 (3p)

Skapa en funktion som tar in två punkter i 2D (p_1, p_2) , (q_1, q_2) och som returnerar tillbaka avståndet mellan dessa. Anropa därefter denna funktion för test

Det finns en textfil som heter testpunkter.txt som ligger i undermappen **testpoints** och innehåller:

```
Punkt 1, punkt 2
(1,2), (0,0)
(2,0), (3,5)
...
(-2,-2), (9,2)
```

Uppgift 6 (VG-nivå) (4p)

I den här uppgiften ska du skapa ett blackjack som är ett kortspel. Förenklat är reglerna för blackjack:

- dealern tar två kort
- dealern visar sitt första kort
- spelaren tar två kort
- spelaren får välja att ta nytt kort eller stanna
- när spelaren stannar räknar man poängen genom:
 - J, K, Q värda 10p
 - A värd 1 eller 11
 - Övriga kort värda deras valör
- om spelarens poäng mindre eller lika med 21, vinner spelaren om hens poäng överstiger dealerns, annars vinner dealern

Uppgift 7 (VG-nivå) (4p)

Skapa en klass som heter NormalDist som följer detta UML:

NormalDist
-mu : float -sigma : float
+__init__(self, mu:float, sigma:float) +pdf(self, x:float) --> float +plot_pdf(self, x:float) --> None +__add__(self, other) --> NormalDist +__repr__(self) --> str

- vid addition av två objekt av typen NormalDist ska dess respektive **mu** och **sigma** adderas.
- formeln för pdf för normaldistribution:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$