

# Prov i Programmering HT24

Ansvarig lärare: Raphael Korsoski

Max 20p. Godkänt 9p. Väl Godkänt 15p.

Datum: 28e juni, 2024

Alla svar skall motiveras för full poäng. Lycka till!

**1** Betrakta följande kod:

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

Beskriv vad koden gör och vad som ritas upp. *3p*

**2** Vad skrivs ut av följande kod? *3p*

```
1 frukter = ["jordgubbe", "vindruva", "grape"]
2 godfrukt = frukter[2]
3 print(godfrukt[-1])
```

**3** Följande program kraschar:

```
1 def print_row(input_array):
2     for i in input_array:
3         print(f"{int(i[0])}: {i[1:]} ")
4
5 A = "1 Alice\n2 Bob"
6 print_row(A)
```

Felet är ett **ValueError** men du har inte tillgång till stack-trace eller felmeddelandet.

*Tips: En sträng är en lista av Char*

a) *1p* Vilken rad är det som orsakar felet?

b) *2p* Ange hur A skall se ut för att producera det väntade resultatet:

1: Alice 2: Bob

**4** Betrakta följande program, där `my_document` är en lokal modul du inte har direkt tillgång till:

```

1 import requests
2 import my_document
3 import datetime.datetime as dt
4
5 class MessageHandler:
6     def MessageHandler(self, source_url):
7         self.source = source_url
8
9     def get_source_document(self):
10        return requests.get(self.source)
11
12 class MessageDispatcher(MessageHandler):
13     def dispatch_message(self, message, field):
14         doc = self.get_source_document()
15         msg = my_document.update(doc, message, field=field)
16         requests.post(self.source_url, msg)
17
18 remote = MessageDispatcher("https://my-web-server")
19 remote.dispatch_message(
20     f"Last updated {dt.now()}",
21     field='timestamp')

```

a) Vad är värdet på fältet `remote.source`?

*2p*

b) Beskriv vad programmets syfte kan tänkas vara.

*2p*

**5** Beskriv skillnaden mellan en `dictionary` och ett `set`.

*3p*

**6** Betrakta följande kod:

```

1 import numpy as np
2 def sums(A, B):
3     out = 0
4     for l, r in zip(A, B):
5         out += l + r
6     return out
7
8 def np_sums(A, B):
9     return np.sum(np.array(A)+np.array(B))

```

Hur påverkas prestandan relativt den andra funktionen om:

a) Om A och B bara har några få element.

*2p*

b) Om A och B har tusentals element.

*2p*