

Løsningsforslag for teori, oppgave 1 og oppgave 4b

Teori

- 1) Hvilken av de følgende påstandene er korrekte?
Tupler er ikke-muterbare
- 2) Hva er hovedoppgaven til en ALU, og hva står forkortelsen for?
Utføre regneoperasjoner. ALU står for Arithmetic Logic Unit.
- 3) Hvilket tall på desimalform representerer det heksadesimale tallet «DDE»?
3550
- 4) Hva står RGB for?
Rød, Grønn og Blå
- 5) Hvor mange symboler kan representeres med 5 bit?
32

Oppgave 1

a)

```
def mystery(a):  
    b = ''  
    for i in range(0, len(a), 2):  
        b += a[i]  
    return b
```

```
print(mystery('pbajnødran'))
```

Hva vil skrives til skjerm når kodesnutten over kjøres?

- ☒ panda
- ☐ pandabjørn
- ☐ bjørn
- ☐ pbajnødran

I kodesnutten med funksjonen mystery over, hvilken variabeltype vil bli returnert?

- ☐ Boolean
- ☐ None
- ☒ String
- ☐ Integer

b)

Hva skrives til skjerm når `nested_loop(3, 4)` kjøres? (Alternativene er vist over mellom kodesnutten med funksjonen `nested_loop` og denne teksten.)

- ☐ Alternativ 1
- ☐ Alternativ 2
- ☒ Alternativ 3
- ☐ Alternativ 4



Hvilken variabeltype returneres når `nested_loop(3, 4)` kjøres, og hvilken verdi vil denne ha?

- ☐ Float, 12.0
- ☐ String, (korrekt svaralternativ på oppgaven over)
- ☐ Boolean, True
- ☒ Integer, 12



c)

```
def secret(g, k):
    res = []
    for i in range(0, min(len(g), len(k))):
        if i % 2 == 0:
            res.append(g[i])
        else:
            res.append(k[i])
    return res

my_res = secret([1, 0, 0, 1, 1, 0], [0, 1, 1, 0, 1, 0, 0, 0, 1])
print(my_res)
```

d)

```
def another_secret(g):
    return str(g).strip('[]').replace(' ', '')

print(another_secret([1, 1, 0, 0, 1, 0]))
```

Hva vil skrives til skjerm når kodesnutten over kjøres?

- ☒ 110010
- ☐ 010011
- ☐ [010011]
- ☐ [110010]



Både 010011 og 110010 er eksempler på binære tall. Hva er verdiene til disse tallene i det desimale tallsystemet?

- ☐ 38 og 100
- ☐ 19 og 100
- ☒ 19 og 50
- ☐ 38 og 50



e)

```
def reverse(lst):
    res = []
    for i in range(len(lst) - 1, -1, -1):
        res.append(lst[i])
    return res
```

f)

Ved kjøring av kodesnutten over, hva skrives til skjerm?

- ☒ ['answer?', 'final', 'Is', 'that', 'your']
- ☐ ['Is', 'that', 'your', 'final', 'answer?']
- ☐ ['your', 'that', 'Is', 'final', 'answer?']
- ☐ ['answer?', 'final', 'your', 'that', 'Is']

Hva gjør funksjonen doing_something(lst)?

- ☒ Tar inn listen lst og returnerer en ny liste med elementene i lst sortert i alfabetisk rekkefølge (A-Å).
- ☐ Tar inn listen lst og returnerer en ny liste med elementene i lst reversert.
- ☐ Tar inn listen lst og returnerer en ny liste med elementene i lst sortert i omvendt alfabetisk rekkefølge (Å-A).
- ☐ Tar inn listen lst og returnerer en kopi av listen. Selve funksjonen gjør mye unødvendig arbeid, og kunne bare ha returnert lst.copy() fra starten av.

I funksjonen doing_something(lst) blir en kopi av listen lst lagret i variabelen copy i kodelinjen copy = lst.copy(). Om man hadde byttet denne linjen med copy = lst, hva ville vært forskjellen på de to funksjonene?

- ☐ Det ville ikke vært noen forskjell. Ingen av versjonene av funksjonen doing_something(lst) ville ha resultert i at listen lst blir endret.
- ☐ Den opprinnelige funksjonen endrer på listen lst (etter kjøring av doing_something(words), ville man endt opp med words = []). Derimot, ville den foreslåtte endringen resultert i at funksjonen ikke lenger endrer på listen lst (dvs. words blir ikke endret etter kjøring av doing_something(words)).
- ☐ Ingen av de andre alternativene stemmer.

- ☒ Den opprinnelige funksjonen endrer ikke på listen lst (dvs. words blir ikke endret etter kjøring av doing_something(words)). Derimot, ville den foreslåtte endringen resultert i at funksjonen endrer på listen lst, siden både variabelen lst og copy da ville ha pekt på samme objekt i minnet (etter kjøring av doing_something(words), ville man endt opp med words = []).

Oppgave 4b)

```
def products(matrix, vector):  
    res = []  
  
    for   
  
        row =   
  
        for   
  
              
  
              
  
    return res
```