Løsningsforslag

Runde 1

1.1 Grunnleggende programmering.

1.1.1

Skriv et program som regner ut hvor mye vekslepenger kunden skal ha tilbake når

```
pris   =  673
betalt = 1000
```

```
1  pris    =  673
2  betalt = 1000
3  vekslepenger = betalt - pris
4  print(vekslepenger)    # 327
```

1.1.2

Skriv et program som regner ut volumet av en sylinder i liter (dm^3) når målene er oppgitt i cm

```
diameter = 10   # cm
høyde    = 15   # cm
```

```
1  diameter   = 10                  # cm
2  høyde      = 15                  # cm
3  radius     = diameter/2
4  grunnflate = 3.14 * radius ** 2
5  volum      = grunnflate * høyde  # kubikk cm
6  volum      = volum / 1000        # kubikk dm eller liter
7  print(volum)                     # 1.1775
```

1.1.3

Utforsk programmet nedenfor ved å skrive inn ulike navn og kjøre programmet. Prøv også med store og små bokstaver samt siffer og spesialtegn i navnene. Skriv ned hva du fant ut.

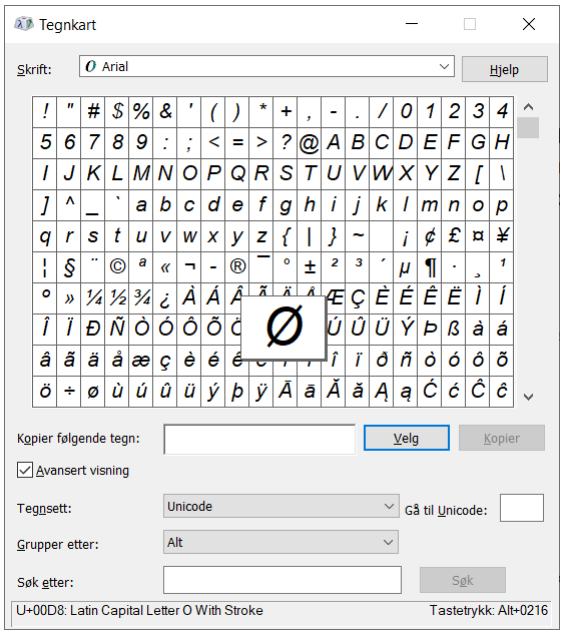
```
1  navn1 = input('Skriv inn det første navnet: ')
2  navn2 = input('Skriv inn det andre navnet: ')
3
4  if navn1 > navn2:
5      print(navn1 + ' er større enn ' + navn2)
6  else:
7      print(navn2 + ' er større eller lik ' + navn1)
```

`Bjarne` er større enn `Arne`. `Arne` er større enn `Anne`. `else` er større enn `Frida`. `2_Arne` er større enn `1_Bjarne`. `Øystein` er større enn `Ægir`, men `Øystein` er også større enn `Åge`!

Tekst(strenger) består bokstaver eller tegn. Hvert tegn har en binær kode med tilhørende tallverdi. Se eksempelvis denne ASCII tabellen. Disse tallverdiene sammenliknes tegn for tegn. Fordi `B` har koden 66 og `A` har koden 65, er `Bjarne` større enn `Arne`. Når vi sammenlikner `Arne` med `Anne`, er det første tegnet det samme så vi må se på det neste tegnet. Fordi `r` har koden 114 og `n` har koden 110, er `Arne` er større enn `Anne`. Små bokstaver har større tallverdi enn store bokstaver, og store bokstaver har større tallverdi enn siffer.

```
>>> import sys
>>> print(sys.getdefaultencoding())
utf-8
>>> print(ord('A'))
65
>>> print(ord('B'))
66
>>> print(ord('Æ'))
198
>>> print(ord('Ø'))
216
>>> print(ord('Å'))
197
```

ASCII tabellen består av 128 tegn som ikke inneholder de norske bokstavene æ, ø, å, Æ, Ø og Å. ASCII tegnene inngår også i tegnsettet UTF-8 som også inneholder de norske bokstavene og er standard i Python. Vi kan finne koden til et tegn med ord (IBF) i Python eller Tegnkart-programmet i Windows. Koden til Å er mindre enn både Æ og Ø så her må vi være på vakt. På et numerisk tastatur kan vi eksempelvis trykke Alt+0216 for å få fram Ø. Uten numerisk tastatur kan vi skrive inn koden i Tegnkart-programmet eller tilsvarende hex kode i Word, merke koden og trykke Alt+X (Det holder med D8).



1.1.4

Bruk eksemplene om beregninger og løkker i teksten til å legge sammen tallene fra 1 til 10 på to forskjellige måter.

```
1  # 1. metode
2  sum1 = 1+2+3+4+5+6+7+8+9+10
3  print(sum1)
4
5  # 2. metode
6  liste = [1,2,3,4,5,6,7,8,9,10]
7  sum2  = 0
8
9  for tall in liste:
10     sum2 = sum2 + tall
11
12 print(sum2)
```


Smidig IT-2

Python har den innbygde funksjonen `range` som kan lage sekvenser av tall. Koden `range(1,11)` lager en sekvens av tallene fra og med 1 til og med 10. `range` kan du bruke på samme måte som en liste i `for`-setningen. Prøv å regne ut den samme summen ved bruk av `range`-funksjonen.

```
14  # 3. metode
15  sum3 = 0
16
17  for tall in range(1,11):
18      sum3 = sum3 + tall
19
20  print(sum3)
```

Python inneholder også den innbygde funksjonen `sum` som kan legge sammen elementer i sekvenser. Prøv å regne ut den samme summen ved bruk av `sum`-funksjonen.

```
22  # 4. metode
23  sum4 = sum(range(1,11))
24  print(sum4)
25
26  # Vi kan også gjøre range om til en liste:
27  print(list(range(1,11)))
```

Vurder de forskjellige metodene ut fra kompetansemål 3[1]. Hvilken metode foretrekker du? Hvordan påvirkes vurderingen dersom det hadde vært tall fra 1 til 10 000?

> Det er alltid flere alternative måter å løse et problem.
>
> Vi kan alltid diskutere fordeler og ulemper med forskjellige alternativer. Hvordan vi koder er også viktig i samarbeid med andre, kompetansemål 9[2]. I The *Zen of Python* finner vi anbefalinger for hvordan vi bør gå fram, eksempelvis *Simple is better than complex* og *Readability counts*. Hele dokumenter ser vi ved å skrive `import this` i fortolkeren.
>
> Det tar tid å programmere, og profesjonelle systemutviklere koster penger. De får gjort jobben fortere dersom de bruker ferdige funksjoner enn å finne opp kruttet på nytt. Dessuten er ferdige funksjoner oftest optimalisert og har få eller ingen feil.
>
> Det er åpenbart at metode 3 og 4 er bedre når det blir mange tall.

Dersom du lærer om rekker i matematikk S2 eller R2, kan du komme tilbake til denne oppgaven senere og løse den på nok en ny måte.

[1] utforske og vurdere alternative løsninger for design og implementering av et program
[2] velge og bruke relevante systemutviklingsmetoder og -verktøy for samarbeid med andre