31.12.2020

Prosjektoppgave

PROG100

Av: Jon Bjarne Bø

Innholdsfortegnelse

Innledning

Utgangspunktet for denne prosjektoppgaven er undervisning i matematikk 2P-Y, altså undervisning for påbyggselever i matematikk. 2P-Y er fremdeles regulert av LK06, men vil forholde seg til ny læreplan våren 2022. Den nye læreplanen spesifiserer at elevene skal programmere i matematikk. Pensum i 2P-Y krever at elevene kan en del om digitale verktøy som Excel og Geogebra. Jeg mener også at programmering kan brukes som et verktøy for å bygge kompetanse hos eleven. I tillegg bidrar programmeringsundervisning til å trene elevene i problemløsning og problemløsningsstrategier som kan tas med inn i matematikkproblemer elevene kan møte.

Denne oppgaven har jeg valgt å dele inn i 2 deler. Den første delen (Undervisningsopplegg 1) er den største delen av oppgaven og inneholder et komplett undervisningsopplegg med en total varighet på omtrent 2 uker (10-12 undervisningstimer). Dette undervisningsopplegget er igjen delt opp i 5 deler. Forutsetningene er at elevene har liten erfaring med programmering fra før, men at elevene har jobbet med grunnleggende forståelse av prosent og renters rente. I tillegg bør de ha arbeidet med figurtall og tallrekker. Tema her er prosent og figurtall (tallrekker) som begge er aktuelle for eksamen for påbyggselevene.

Den andre delen (Undervisningsopplegg 2) er et tredelt undervisningsopplegg tenkt å vare omtrent 4-6 skoletimer hvor elevene har vært gjennom begrepene som presenteres i undervisningsopplegg 1. De har altså hatt litt programmering fra tidligere, og de har jobbet godt med oppgavene og vært til stede i gjennomgangene. Denne økten har statistikk som tema, og krever at elevene har litt forkunnskaper om noen generelle statistiske begreper som gjennomsnitt og median. Den siste delen i dette opplegget er en større oppgave som elevene skal levere inn.

De mest grunnleggende programmene som kommer i starten av undervisningsopplegg 1 er ikke lagt ved som Python-file, dette er avklart med Finn. Resten av programmene ligger som .py-filer i zip-mappen hvor denne oppgaven ligger. Python-filene har fått navn etter undervisningsopplegg, del og oppgavenummer. En fil knyttet til undervisningsopplegg 2, del 2 og oppgavenummer 4 vil derfor ha navn U2D2O4.

Undervisningsoppleggene er strukturert slik at den kursiverte teksten er kommentarer og forklaringer til læreren som gjennomfører undervisningsoppleggene, mens den teksten som ikke er kursivert er det elevene får utdelt. Dokumentet med oppgaver til elevene ligger vedlagt som «Elevoppgaver.docx» i zip-mappen. Eventuelle Excel-filer som brukes ligger også i samme mappe.

Undervisningsopplegg 1 - Programmering med prosent og figurtall

Utførelse:

Kursiverte instrukser er til læreren, skrift uten formatering er direkteinstrukser til elevene.

Del 1 – Sentrale programmeringsbegreper – variabler og aritmetikk

Gjennomgang av lærer, dialog i klasserommet og elevoppgaver

Elevene får utdelt et hjelpehefte til Python som jeg har laget selv (dette ligger vedlagt og er kalt «Introduksjon til programmering i Python») i tillegg til dette kommandokartet fra UiO.

Lærer starter økten med å åpne et program som ser slik ut, og lar elevene komme med forslag til hvilken verdi variabelen c inneholder.

```
1  a = 25
2  b = 20
3
4  c = a + b
5
```

Kjører programmet og ser at ingenting skjer. Vi kommer da inn på print-funksjonen og hvorfor vi må bruke denne om vi vil se hvilken verdi variabelen c har.

Etter dette kan læreren endre variablene slik at elevene får se eksempler på hva som skjer om man benytter seg av både strenger og heltall sammen:

```
1  a = "Hei"
2  b = 20
3
4  c = a + b
5
6  print (c)
```

```
1  a = "Hei"
2  b = 20
3
4  c = a * b
5
6  print (c)
```

Dette blir en fin overgang til å snakke om de ulike datatypene (som også står beskrevet på arket som elevene fikk utdelt). Før elevene jobber med oppgaver viser læreren hvordan man formaterer tekst og variabler i print-funksjonen slik som bildet under. Om ønskelig kan læreren også nevne input-funksjonen og vise noen eksempler med dette.

```
1  a = 5
2  b = 2
3  c = a * b
4
5  print(f'{a} * {b} = {c}')
```

Hvis tid kan man vise hva som skjer om man summerer et heltall og et flyttall, eller to strenger, men dette kan også elevene teste litt ut på egenhånd gjennom oppgavene under.

Elevoppgaver

Uthevet tekst er tekst som skal byttes ut med verdier som vi har lagret i variabler tidligere i programmet.

Oppgave 1:

Lag et program som lagrer navnet ditt i en variabel og skriver ut «Hei, **navnet ditt**!».

Oppgave 2:

Utvid programmet fra forrige oppgave med en variabel som inneholder alderen din. Nå skal du få programmet til å skrive ut «Hei, **navnet ditt**! Du er **alderen din** år».

Oppgave 3:

Lag et program med tre variabler a, b og c. Bestem en verdi for variabelen a. Variabelen b skal være dobbelt så stor pluss 5 som a, og variabelen c skal være halvparten så stor som b. La programmet skrive ut verdiene til variablene.

Oppgave 4:

Arne har laget et program som skal regne ut hvor mye penger han har på kontoen etter 1 år med renter. Han får en rente på 3,5% og setter inn 25 000 kroner. Programmet han har laget ser slik ut:

```
prosent = 3.5
prosentfaktor = 3,5/100
vekstfaktor = 1 + 3,5/100

pengerPåKontoenFørRenter = 25000
pengerPåKontoenEtterRenter = 25000 * vekstfaktor

print (Etter ett år med renter har du pengerPåKontoenEtterRenter på kontoen din!)
```

Det er noen feil i programmet til Arne som gjør at han får opp feilmeldinger når han kjører programmet. Feilsøk programmet, rett opp i feilene til Arne, og skriv inn riktig kode i boksen under.

Oppgave 5:

Bruk koden som du laget i forrige oppgave til å finne ut hvor mye en vare som opprinnelig koster 350 kroner koster, om den er satt ned med 34%.

Del 2 – Løkker (45-90 min)

Gjennomgang av lærer, dialog i klasserommet og elevoppgaver

Om det tidligere er blitt arbeidet med trekanttall, kan man trekke paralleller til dette (finne trekanttall nummer 100 numerisk), hvis ikke kan oppgaven som skal gjennomgås være å summere alle tallene fra 1 til 100. Videre tar jeg utgangspunkt i at elevene har arbeidet med trekanttall på forhånd.

Oppgave 1 (Felles):

Felles: Lag et program som skriver ut de første 20 heltallene. *Gi oppgaven til elevene først, og gå gjennom etterpå. Før du viser med kode, forsøk å benytte pseudokode. Vis både med while-løkker og for-løkker. Hva er forskjellen på de to programmene under?:*

```
1  a = 0
2
3  while a <= 20:
4      print (a)
5      a += 1
```

```
1  a = 0
2
3  for i in range (20):
4      a += 1
5      print (a)
```

Åpne programmet som viser under og la klassen diskutere hva programmet gjør. Hvordan kan vi endre på programmet om vi vil ha med alle partallene opp til og med 100? La elevene forsøke å lage det samme programmet ved å bruke en for-løkke.

```
1  a = 0
2
3  while a < 100:
4      print (a)
5      a += 2
```

Oppgave 2 (Felles):

Lag et program som skriver ut trekanttall nummer 100. *(Her vil muligens mange av elevene bruke formelen for trekanttall nummer n, men da kan man trekke paralleller til dette senere)*

Når elevene har fått prøve seg litt her gjennomgår man det felles. Her kan man starte med å vise hvordan man kan bruke formelen for trekanttall slik:

```
n = 100
trekanttall = int(n*(n+1)/2)

print(f'Trekanttall nummer {n} er  {trekanttall}')
```

Når læreren har snakket litt om dette kan vi trekke inn numeriske metoder. Hvordan kunne vi gjort dette om vi ikke hadde hatt formelen? Man kan etter litt dialog med klassen da introdusere løkker, og vise hvordan man kan regne ut trekanttall nummer 100 numerisk enten med en for-løkke:

```
a = 1
n = 100
trekanttall = 0

for i in range (n):
    trekanttall = trekanttall + a
    a += 1

print(f'Trekanttall nummer {n} er  {trekanttall}')
```

Eller med en while-løkke:

```
a = 1
n = 100
trekanttall = 0

while a <= n:
    trekanttall = trekanttall + a
    a += 1

print(f'Trekanttall nummer {n} er  {trekanttall}')
```

Elevoppgaver

Om du ikke får til syntaksen så forsøk å skrive pseudokode først

Oppgave 3:

Hva er forskjellen på de to programmene under? Skriv opp hva som blir skrevet ut i de to ulike programmene uten å skrive inn koden selv.

```
1  a = 0
2
3  for i in range (20):
4      a += 1
5  print (a)
```

Hva skriver programmet ut her?

```
1  b = 0
2
3  for i in range (20):
4      b += 1
5      print (b)
```

Hva skriver programmet ut her?

Oppgave 4:

Lag et program som skriver ut de første 15 kvadrattallene.

Oppgave 5:

Tallfølgen under består av uendelig mange tall som følger et fast mønster. Lag et program som regner ut tall nummer 60 i tallrekken ved å bruke løkker.

$$3, 7, 11, 15, 19, 23, \ldots$$

Oppgave 6:

Du setter inn 100 000 kroner på en sparekonto med 3,2% rente. Lag et program som skriver ut hvor mye du har på kontoen hver av de 10 første årene om pengene står urørt. Du skal løse denne oppgaven ved å bruke løkker.

Oppgave 7:

Du starter å spare på BSU. Du setter inn 25000 kroner 1. Januar hvert år. Den årlige renten på BSU-kontoen er 3,5%. Lag et program som skriver ut hvor mye du har på kontoen hver av de 10 første årene. Husk at du setter inn 25 000 hvert eneste år.

Del 3 – Suksessive renteberegninger, figurtall og programmering

Elevoppgaver

Lærer er selvfølgelig til stede i arbeid med disse oppgavene. Mange av oppgavene her kan oppfattes som utfordrende, og det er mulig at noen av oppgavene må jobbes med i fellesskap.

Dere skal denne økten arbeide med figurtall og forsøke å løse oppgaver ved å bruke numeriske metoder på rekursive formler (altså ved å finne et mønster til en figur og de neste figurene i rekken). Dere skal bruke programmering som et verktøy for å finne figurtall langt bak i rekken, og programmering skal kunne fungere som et verktøy for dere om dere ikke klarer å komme frem til den eksplisitte formelen. Noen av oppgavene dere her skal jobbe med er hentet fra tidligere prøver.

Oppgave 1

For 10 år siden arvet du 140 000 kroner av din bestemor. Disse pengene ble satt inn på en konto, som har en årlig rente på 2,1%. Du skal nå bruke disse pengene, og du tar ut 10 000 kroner årlig. Lag et program som skriver ut hvor mye det står igjen på kontoen hvert år de neste 20 årene. Hvor mange år tar det før kontoen er tom?

Oppgave 2

Fra 2009 til 2014 var maksimalt innskudd på BSU 20 000 kroner. Fra og med 2014 har det maksimale innskuddet på BSU-kontoer vært 25 000 kroner. Arne satte penger inn på BSU-kontoen sin i 2009. Vi antar at den årlige renten har vært konstant på 4,1%. Lag et program som skriver ut hvor mye Arne har på kontoen i år om han har satt inn maksimalt beløp hvert år.

Oppgave 3

Fibonacci-tallene er en berømt tallfølge som finnes flere steder i naturen. Tallfølgen er bygd opp slik at hvert tall i følgen er summen av de to foregående tallene i følgen. De første tallene i følgen er:

$$1, 1, 2, 3, 5, 8, 13, 21, \ldots$$

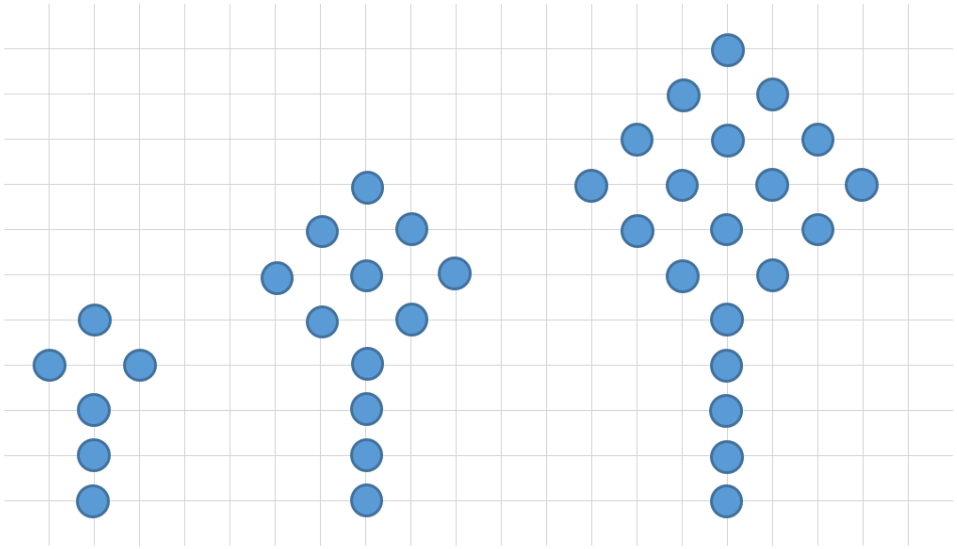
Lag et program som skriver ut de første 50 Fibonacci-tallene.

Oppgave 4

Ta utgangspunktet i tallrekken fra oppgave 6 (Fibonacci-tallene) og lag et program som regner ut **summen** av de første 60 Fibonacci-tallene.

Oppgave 5

Nedenfor ser du tre figurer. Tenk deg at du skal fortsette å lage figurer etter samme mønster.



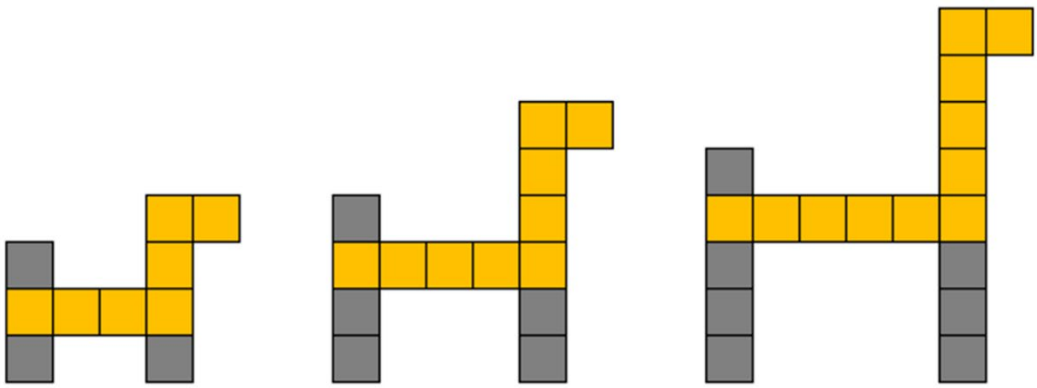
Figur 1 Figur 2 Figur 3

a) Hvor mange prikker vil det være på figur nummer 5?

b) Lag et program som regner ut hvor mange prikker det er i figur nummer 100 og skriver svaret til skjerm.

Oppgave 6

Nedenfor ser du tre figurer satt sammen av små kvadratiske klosser. Tenk deg at du skal fortsette å lage figurer etter samme mønster.



Figur 1 Figur 2 Figur 3

a) Hvor mange **gule** kvadrater trenger du for å lage figur nummer 43?

b) Hvor mange kvadrater trenger du for å lage figur nummer 43?

Du skal bygge så mange av figurene over som mulig.

c) Hvor mange klosser trenger du for å lage de første 20 figurene (altså summen av de første 20).

d) Bruk en while-løkke til å regne ut hvor mange figurer du klarer å lage om du har 558 klosser totalt og vil lage så mange ulike figurer som mulig.

Oppgave 7

Du er på hytta og bestemmer deg for å bygge korthus. Under ser du de tre første korthusene i en serie som kan fortsettes etter samme mønster. Hver blå strek representerer et kort.

Korthus 1 Korthus 2 Korthus 3

a) Hvor mange kort trenger man for å lage korthus nummer 23?

I en kortstokk er det 52 vanlige kort i tillegg til 2 jokere, altså 54 kort totalt.

b) Lag et program som hjelper deg å finne ut hvilken figur som er den største du kan lage med 5 slike kortstokker. Programmet skal også skrive ut hvor mange kort du da har til overs.

Del 4 – Funksjoner

Etter at elevene har fått sett på oppgave 6 (og kanskje 7) tar læreren en gjennomgang av dette på tavlen. Programmet for oppgave 6 kan se omtrent slik ut:

```
a = 0
b = 1
antall = 50 #Antall fibonacci-tall

for i in range(antall):
    a += b
    print(a)
    b = a - b
```

Etter at læreren har gjennomgått dette og elevene er enige i oppbygningen av programmet kan man gå videre for å introdusere funksjoner. Læreren forklarer hvordan funksjoner i Python fungerer og hvorfor det kan være nyttig. Etter dette laget læreren en funksjon som skriver ut de første x Fibonacci-tallene med utgangspunkt i programmet over. Programmet vil se slik ut som vist i oppgave 1 under. Etter dette skal elevene gjøre oppgave 1.

Elevoppgaver

Oppgave 1

```
def f(n):
    a = 0
    b = 1
    antall = n #Antall fibonacci-tall

    for i in range(antall):
        a += b
        print(a)
        b = a - b

f(60)
```

Gjør endringer i programmet over slik at funksjonen kun skriver ut det Fibonaccitallet som en er ute etter, f.eks. f(3) skriver ut 2.

Oppgave 2:

Lag en funksjon som skriver ut partall nummer n, der n er input i funksjonen. La programmet skrive ut partall nummer 50, 100 og 1024.

Oppgave 3:

Lag en funksjon som skriver ut summen av de første n kvadrattallene, der n er input i funksjonen. La programmet skrive ut summen av de første 20 kvadrattallene.

Oppgave 4:

Programmet under inneholder en funksjon som regner ut partall nummer n. Lag en løkke som inneholder denne funksjonen slik at du kan finne summen av de første 100 partallene.

```
1 def partall(n):
2     p = 2 * n
3     return p
4
5 print(partall(5))
```

Her er tanken at elevene skal se at de kan bruke funksjoner til noe annet enn å skrive til skjerm og noe annet enn å kalle direkte på funksjonen én gang. Programmet bør gjennomgås av lærer etter at elevene har jobbet litt med dette og kan se ut som programmet under. Lærer bør også snakke om årsaken til at det står «return» i bunnen og la elevene se forskjellen på om de har dette med eller ikke i programmet.

```
1 summ = 0
2 def partall(n):
3     p = 2 * n
4     return p
5
6 for i in range(100):
7     summ += partall(i+1)
8
9 print(summ)
```

Her kan læreren også vise at man kan lage to funksjoner og sette dem sammen slik for å få samme resultat:

```
1  def partall(n):
2      p = 2 * n
3      return p
4
5  def summ(n):
6      s = 0
7      for i in range(n):
8          s += partall(i+1)
9      return s
10
11 print(summ(100))
```

Del 5 – Array

Her er tanken at elevene i forkant har jobbet med de forrige delene. De skal i tillegg ha sett noen videoer i forkant av denne økten. Enten som lærer har laget selv, eller noe liknende dette som ligget på nett (Alle disse videoen er fra samme video, men det kan være fordelaktig å dele opp, gjerne med en oppgave eller to etter hver videosnutt):

Hvordan lage array (dette med datatyper er ikke like relevant, så her ville jeg laget noe selv heller): https://www.youtube.com/watch?v=QUT1VHiLmmI&t=668s

Hente og endre spesifikke elementer i et array, rekker, kolonner etc.:
https://www.youtube.com/watch?v=QUT1VHiLmmI&t=968s

Lage arrayer med kun 0, 1, fulle, random etc.:
https://www.youtube.com/watch?v=QUT1VHiLmmI&t=1394s

Når elevene har sett over dette og har litt bakgrunnskunnskaper om array så kan vi implementere dette i oppgaver som vi tidligere har arbeidet med.

Læreren starter økten med å ta opp følgende eksempel på tavlen og gå gjennom programmet med elevene. Hva gjør programmet og hva blir skrevet ut?:

```
1  import numpy as np
2
3  array = np.zeros(10)
4
5  array[0] = 5
6
7  print(array)
```

Etter en diskusjon om programmet går lærer tilbake til noe som er blitt jobbet med tidligere for å rette arrayer mot matematikken. Starter med å vise programmet under på prosjektoren:

```
1  import numpy as np
2
3  def partall(n):
4      p = 2 * n
5      return p
6
7  array = np.zeros(10)
8
9
10 print(array)
```

Tanken nå er at elevene skal bli utfordret til å lage en array med de første 10 partallene med utgangspunkt i linjene som står over. Elevene arbeider med dette og lærer veileder før det gjennomgås på tavlen igjen og man ender opp med et program som likner på det under.

```python
import numpy as np

def partall(n):
    p = 2 * n
    return p

array = np.zeros(10)

for i in range(10):
    array[i] = partall(i+1)

print(array)
```

Elevene skal nå trene på å manipulere arrayer ved å jobbe med oppgavene under mens lærer går rundt og veileder. Oppgavene står på neste side.

Elevoppgaver

Oppgave 1:

a) Lag en array med de 20 første heltallene ved å bruke np.zeros og en for-løkke.

b) Ta utgangspunktet i arrayen du laget i forrige oppgave og endre denne slik at den inneholder de 20 første partallene.

c) Manipuler arrayen slik at den inneholder de 20 første tallene i 7-gangen.

d) Manipuler arrayen en siste gang slik at den inneholder alle heltallene fra 11 til 30.

Oppgave 2:

Lag en array som inneholder de første 100 tallene i 6-gangen. Bruk denne arrayen til å finne summen av de første 100 tallene i 6-gangen.

Oppgave 3:

Lag en funksjon som lager en array som inneholder de n første Fibonacci-tallene, der n er input i funksjonen. Programmet skal deretter bruke denne funksjonen til å regne ut summen av de første 30 Fibonnaci-tallene.

Når elevene har kommet til den siste oppgaven kan det godt løftes opp på tavlen og man kan kommentere og reflektere litt over funksjonen i programmet og hvordan man kan bruke numpy til å enkelt regne ut summer ved å bruke np.sum. Programmet kan se omtrent slik ut:

```
1  import numpy as np
2
3  #Funksjon som lager array av n fibonacci-tall
4  def f(n):
5      a = 0
6      b = 1
7      array = np.zeros(n)
8
9      for i in range(n):
10         a += b
11         b = a - b
12         array[i] = a
13     return array
14
15 print(np.sum(f(30)))
```

Undervisningsopplegg 2 – Programmering i statistikk

Utførelse:

Kursiverte instrukser er til læreren, skrift uten formatering er direkteinstrukser til elevene.

Del 1 – Lage diagrammer med pyplot, og lagre diagrammene som bildefiler

Den første delen av denne økten er en innføring i matplotlib.pyplot-biblioteket i Python.

Elevene får kopi av programmet under, og får i oppgave å finne ut hva som gjør hva i programmet og legge inn kommentarer. Etter at elevene har fått testet programmet litt og forsøkt å legge inn kommentarer, går læreren gjennom eksempelet. Programmet lager ulike diagrammer med utgangspunkt i karakterfordelingen på en terminprøve, og viser de vanligste diagramtypene som er aktuelle for elevene å bruke.

```
import matplotlib.pyplot as plt
import numpy as np

karakter = np.array([1, 2, 3, 4, 5, 6])
frekvens = np.array([2, 5, 8, 7, 6, 1])

fig1 = plt.figure(1, figsize=(20, 10))
plt.bar(karakter, frekvens, width = 0.8, alpha=0.9)
plt.title('Karakterfordeling terminprøve 2P-Y')
plt.xlabel('Karakter')
plt.ylabel('Antall elever')
plt.show()

fig2 = plt.figure(2, figsize=(20, 10))
plt.pie(frekvens, labels=karakter, autopct='%.1f%%')
plt.title('Karakterfordeling terminprøve 2P-Y')
plt.show()

fig3 = plt.figure(3, figsize=(20, 10))
plt.plot(karakter, frekvens, 'bo')
plt.title('Karakterfordeling terminprøve 2P-Y')
plt.show()

fig1.savefig("Stolpediagram.png")
```

Lærer snakker også om dette med lagring av filer og hvordan man kan endre på filtypen i programmet for å lage ulike bildefiler. Etter at elevene har jobbet med programmet og læreren har gjennomgått dette skal elevene jobbe med oppgavene under. Oppgavene innebærer konstruksjon av diagrammer i tillegg til manipulering av arrayer (som ble arbeidet med forrige runde). Oppgavene krever litt av elevene, og lærer må være tilgjengelig for veiledning. Lærer bør også vise hvordan man importerer math-biblioteket og hvordan man regner ut kvadratrot (siden elevene skal regne ut standardavvik under).

Elevoppgaver

Oppgave 1 (Hentet fra eksamen 2P-Y, H2017 oppgave 4 del 1)

Et idrettslag har 240 medlemmer. Idrettslaget har fire forskjellige aktivitetsgrupper.

Medlemmene fordeler seg slik:

Aktivitetsgruppe	Antall medlemmer
Langrenn	60
Hopp	40
Freestyle	80
Alpint	60

Lag et program som lager et sektordiagram som fremstiller dataene i tabellen over på en oversiktlig måte.

Oppgave 2

I en klasse er det 16 elever. Tabellen nedenfor viser hvor mange søsken de 16 elevene har.

Antall søsken	Frekvens
0	5
1	6
2	2
3	2
4	1

a) Lag en array for antall søsken og en array for frekvensene. Bruk de to arrayene til å regne ut gjennomsnittlig antall søsken.

b) Bruk arrayene fra forrige oppgave til å regne ut standardavviket for antall søsken.

c) Lag et program som lager et sektordiagram. Diagrammet skal presentere dataene i tabellen oversiktlig, og inneholde overskrift i tillegg til en oversikt over hvor stod andel som er i hver kategori. Programmet skal også lagre diagrammet som en .svg-fil.

Oppgave 3

Tabellen nedenfor viser karakterfordelingen ved en skole ved norskeksamen våren 2017.

Karakter	Antall elever
1	3
2	12
3	25
4	12
5	6
6	2

a) Lag en funksjon som regner ut gjennomsnittet ved å te to arrayer som input, lag så en array for karakterene og en array for antall elever, sett arrayene inn i funksjonen og la programmet regne ut gjennomsnittskarakteren.

b) Utvid programmet slik at det regner ut standardavviket.

c) Utvid programmet ytterliggere slik at det lager et sektordiagram og et stolpediagram. Diagrammene skal fremstille dataene fra tabellen over på en oversiktlig måte. Begrunn hvilket diagram du mener er best egnet til å presentere dataene.

Del 2 – Lese Excel-filer ved å bruke pandas-biblioteket

Denne delen er siste veldig kort innføring i pandas slik at elevene kan importere data fra Excel-filer som de kan manipulere.

Lærer starter med å vise et eksempel hvor en importerer data fra en Excel-fil. Programmet og utklipp fra Excelfilen vises under.

Excelfil (skjermutklipp med alt innhold):

	A	B	C	D
1	Karakter	4PÅBA	4PÅBB	4PÅBC
2	1	4	2	1
3	2	5	8	4
4	3	7	5	8
5	4	6	7	8
6	5	4	4	4
7	6	1	1	3
8				

Karakterfordelingseksempel.xlsx

Programkode som gjennomgås på tavla:

```
import pandas as pd

fil = 'Karakterfordelingeksempel.xlsx'
data = pd.read_excel(fil)

karakter = data['Karakter'].values
klasseA = data['4PÅBA'].values
klasseB = data['4PÅBB'].values
klasseC = data['4PÅBC'].values

print(karakter)
print(klasseA)
print(klasseB)
print(klasseC)
```

Lærer snakker her kort om hvordan dataen fra Excel-filene nå er omgjort til arrayer som kan behandles slik de er kjent med fra før. Etter dette får elevene i oppgave å bruke det de har lært tidligere til å regne ut antall elever i hver klasse i tillegg til gjennomsnittskarakterer i de tre klassene fra eksempelet som lærer viste. Elevene kan også få i oppgave å regne ut det samme i Excel for å kontrollere at svaret som programmet gir er riktig.

Programmet som elevene kommer frem til kan se omtrent slik ut:

```
import pandas as pd
import numpy as np

fil = 'Karakterfordelingeksempel.xlsx'
data = pd.read_excel(fil)

karakter = data['Karakter'].values
klasseA = data['4PÅBA'].values
klasseB = data['4PÅBB'].values
klasseC = data['4PÅBC'].values

eleverA = np.sum(klasseA)
eleverB = np.sum(klasseB)
eleverC = np.sum(klasseC)

gjA = np.sum(karakter*klasseA)/eleverA
gjB = np.sum(karakter*klasseB)/eleverB
gjC = np.sum(karakter*klasseC)/eleverC

print(f'I 4PÅBA er det {eleverA} elever og gjennomsnittskarakteren er {gjA:.2f}.')
print(f'I 4PÅBB er det {eleverB} elever og gjennomsnittskarakteren er {gjB:.2f}.')
print(f'I 4PÅBC er det {eleverC} elever og gjennomsnittskarakteren er {gjC:.2f}.')
```

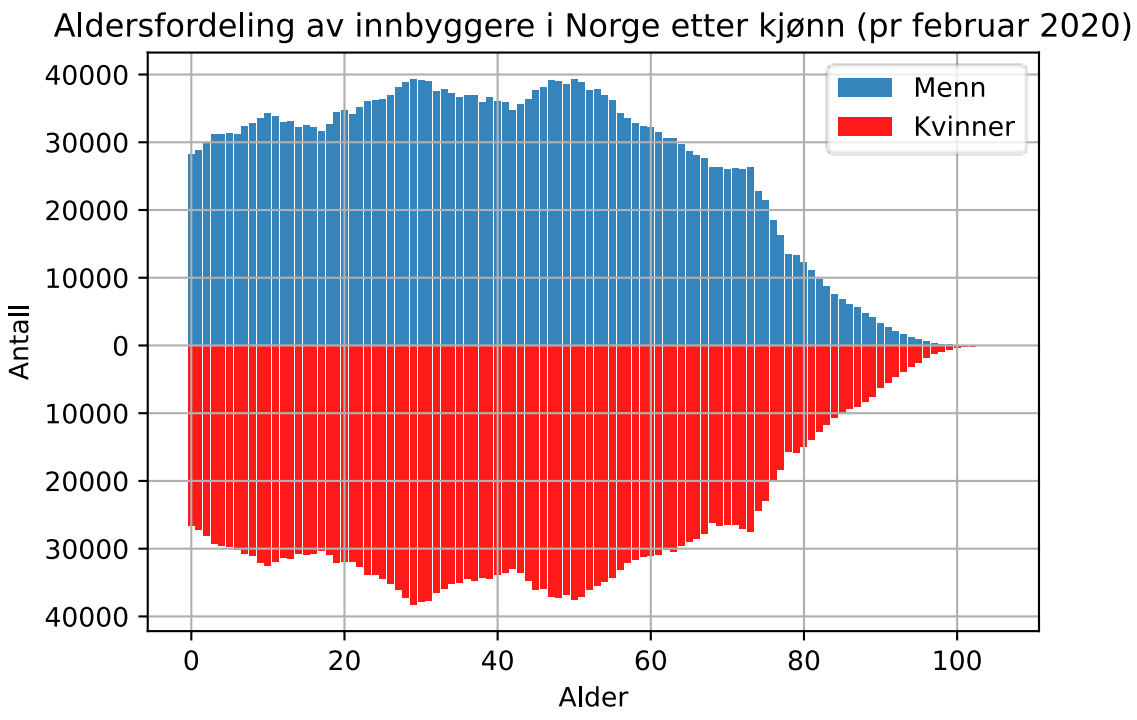
Del 3 – Sammensatt innleveringsoppgave

Avslutningsvis for å teste elevenes forståelse knyttet til det meste av det som er blitt gjennomgått får elevene i oppgave å ta lage et program som gjør statistiske beregninger med utgangspunkt i data fra en Excelfil som lærer legger ut. Oppgaven elevene får står under:

Innleveringsoppgave:

Ta utgangspunkt i Excelfilen kalt «Aldersfordeling norge.xlsx». Du skal lage et program som henter inn data fra dette Exceldokumentet og tilfredsstiller punktene under.

- Programmet skal regne ut gjennomsnittsalder i Norge både for menn og for kvinner og skrive disse til skjerm. I tillegg skal programmet regne ut den totale gjennomsnittsalderen i Norge og skrive denne til skjerm.
- Programmet skal regne ut standardavviket for alderen både for menn og kvinner (her trenger du ikke å regne ut det sammenlagte standardavviket) og skrive denne til skjerm.
- Programmet skal lage et diagram som du mener er best mulig egnet til å fremstille dataene fra Exceldokumentet. Om du ønsker en utfordring kan du forsøke å få programmet til å lage et diagram slik som det under:



- Programmet skal skrive det diagrammet du velger til enten en bildefil eller til en PDF-fil.

Ønsker du en ekstra utfordring, eller blir ferdig tidlig kan du forsøke å lage programmet slik at det også kommer tekst under diagrammet i PDF-filen som besvarer kulepunktene.

Eksempel på program som tilfredsstiller oppgaven over ligger vedlagt som U2D3.py.

Vurderingskriterier for opplegget (De grønne rubrikkene er spesifikke i programmering, men er ikke kompetansemål i læreplanen i dag):

Kompetansemål i matematikk (kun punkter som er relevante for opplegget er tatt med)	Høy måloppnåelse	Middels måloppnåelse	Lav måloppnåelse
Eleven skal regne med prosent og vekstfaktor, gjøre suksessive renteberegninger og regne praktiske oppgaver med eksponentiell vekst.	Kan i tillegg gjøre suksessive renteberegninger for å finne en gammel verdi om man har et resultat av en renteendring og suksessiv renteendring. Kan også løse sammensatte praktiske oppgaver.	Kan i tillegg finne ut hvor stor prosentandel en verdi er av en annen, og bruke vekstfaktor for å regne seg frem til nye verdier, og gjøre suksessive renteberegninger ved vekst og nedgang.	Kan regne ut prosentdelen av et tall, og bruke dette for å finne nye verdier om noe øker eller synker.
Eleven skal analysere praktiske problemstillinger knyttet til dagligliv, økonomi, statistikk og geometri, finne mønster og struktur i ulike situasjoner og beskrive sammenhenger mellom størrelser ved hjelp av matematiske modeller.	Kan finne og begrunne mønster i figurer og tallrekker og forklare hvordan disse vokser enten ved å se på figurene eller tallrekken. Kan også komme frem til en rekursiv og en eksplisitt formel for figur n.	Kan finne mønster i figurer og tallrekker og forklare hvordan disse vokser enten ved å se på figurene eller tallrekken. Kan også komme frem til en rekursiv formel som forklarer dette mønsteret.	Kan finne og forklare mønster om en tallrekke vokser jevnt.
Eleven skal kunne beregne og drøfte sentralmål og spredningsmål.	Kan i tillegg regne ut standardavvik når data er presentert i frekvenstabell.	Kan i tillegg regne ut gjennomsnitt for data som er listet opp i frekvenstabeller. Eleven kan også regne ut standardavvik for opplistet data.	Kan finne gjennomsnitt for et datasett når all data er listet opp.
Eleven skal kunne representere data i tabeller og diagrammer og drøfte ulike datafremstillinger og hvilke inntrykk de kan gi.	Kan lage oversiktlige histogram i tillegg til sektor-, stolpe-, linjediagram med tydelige forklaringer og overskrifter. Eleven reflekterer også godt rundt bruken av diagrammer og hvilke diagram som bør brukes i ulike settinger. Eleven reflekterer også rundt hvordan diagrammer fremstilt i media kan virke misvisende.	Kan lage oversiktlige sektor-, stolpe-, og linjediagram med forklaringer og overskrift, men mangler refleksjon rundt hvilke diagrammer som er hensiktsmessig når.	Kan lage enkle diagram, men diagrammene mangler forklaringer og/eller kunne blitt fremstilt på bedre måter.

Mål i programmering	Høy måloppnåelse	Middels måloppnåelse	Lav måloppnåelse
Eleven skal kunne bruke variabler, operatorer og løkker i Python til å løse ulike matematikkproblemer.	Viser forståelse for begrepene variabler og løkker, og kan bruke løkker, variabler og operatorer til å løse problemer. Kan også skifte mellom både for- og while-løkker, og bruke flere løkker i hverandre.	Viser noe kunnskap om variabler og løkker, og kan bruke variabler, operatorer og løkker til å løse ulike matematikkoppgaver.	Viser manglende kunnskap om variabler og løkker. Kan løse enkle oppgaver som innebærer variabler.
Eleven skal kunne bruke og definere egne funksjoner i Python til å løse ulike matematikkproblemer.	Kan bruke innebygde funksjoner i tillegg til å definere større funksjoner selv. Funksjonene som defineres kan inneholde return og kan brukes til ulike sammensatte oppgaver.	Kan bruke flere innebygde funksjoner i tillegg til å kunne definere enkle funksjoner som skriver ut verdier.	Kan bruke enkle innebygde funksjoner som print og round.
Eleven skal kunne bruke ulike tilleggspakker i Python som math, numpy, pandas og matplotlib.pyplot til å løse ulike matematikkproblemer.	Kan bruke funksjonene i math- numpy- pandas- og pyplot-bibliotekene til å gjennomføre alle beregninger som kreves for å løse problemet. Eleven kan også finjustere input i disse bibliotekene slik at tekst og utregninger ser oversiktlig ut.	Kan bruke noen funksjoner i math- numpy- pandas- og pyplot-bibliotekene til å gjennomføre beregninger.	Kan bruke enkle operasjoner i math-biblioteket eller numpy-biblioteket for å gjennomføre utregninger.
Struktur	Har ypperlig struktur. Koden er lett å lese og inneholder gode forklaringer og inndelinger. Variablene har også fått hensiktsmessige navn.	Har god struktur, men koden opplever som «klumpete» noen ganger. Kunne også med fordel inneholdt flere kommentarer.	Har liten struktur, koden er vanskelig å lese og inneholder få kommentarer