

Det Matematisk-naturvitenskapelige fakultet
UNIVERSITETET I BERGEN
Eksamen i emnet INF 109 – Programmering for naturvitenskap,
Tirsdag 25.november 2014, kl. 09-12

Alle skrevne og trykte hjelpemidler er tillatt.

Oppgavesettet består av **fem** sider med til sammen **tre** oppgaver. Hver oppgave har en prosentvekt, og du bør bruke denne vekten som en rettesnor for hvor mye tid du skal bruke på hver oppgave: Husk at det er lurt å gjøre alle oppgavene.

Du vil få bruk for noen filer som ligger på «MyDocuments». Dette gjelder

- `graphics.py` som er grafikkbiblioteket du er vant med å bruke,
- `us-temperatur.csv` som er en datafil du skal bruke i oppgave 3,
- `EksamenINF109-H2014.py` som er en programfil du skal bruke i oppgave 3, og som du kan bruke som mal for hele besvarelsen din. Du kan åpne denne filen ved først å starte IDLE (på skrivebordet) og deretter åpne filen.

Lag én programfil som inneholder hele besvarelsen, med navn

`EksamenINF109-H2014-<dittnavn>.py`.

Sett inn kommentarer som gjør det klart hvilken oppgave du adresserer, slik:

Løsning av Oppgave 1

<Programkode for Oppgave 1>

Løsning av Oppgave 2

<Programkode for Oppgave 2>

Løsning av Oppgave 3

<Programkode for Oppgave 3>

Du kan bruke filen `EksamenINF109-H2014.py` som mal for besvarelsen din.

Denne filen har allerede strukturen skissert ovenfor.

Skriv forøvrig fornuftige kommentarer i programfilene dine der det er nødvendig, slik at det går klart frem hvordan du har tenkt.

Når du er ferdig, laster du opp besvarelsen i ved å klikke på «Innlevering/Submission» på skrivebordet og følge instruksjonene der.

Oppgave 1 (15%)

Tverrsummen til et heltall er summen av sifrene i tallet. F.eks. er tverrsummen til 9876543210 lik $9+8+7+6+5+4+3+2+1+0 = 45$. Når tverrsummen selv er et tall med flere sifre, kan vi ta tverrsummen av tverrsummen; tverrsummen av 45 er lik 9. Holder vi på slik inntil tverrsummen blir et ensifret tall, har vi funnet *minste tverrsum* for det opprinnelige tallet. F.eks. er minste tverrsum til 9876543210 lik 9. Skriv et Python-program som ber brukeren om et heltall, og deretter skriver ut minste tverrsum til tallet. Lag og bruk hjelpefunksjoner om nødvendig.

Oppgave 2 (25%)

- a) (10%) Lag en metode/funksjon `lesTallListe(Ledestreng)`. Denne tar en innparameter `Ledestreng`, som forutsettes å være en tegnstreng. Metoden skal skrive ut strengen

```
"Skriv inn en liste av "+Ledestreng+"tall adskilt med blanke: "
```

Brukeren skal så taste inn en slik liste, for eksempel

```
6 8 13 12 43 27 5 28 1
```

Deretter skal metoden *returnere* en *liste av tallverdier*, i eksempelet vil det si listen

```
[6, 8, 13, 12, 43, 27, 5, 28, 1]
```

- b) (15%) Skriv en Python-funksjon `twoGreatest(innliste)`. Funksjonen tar en innparameter `innliste`, som forutsettes å være en liste av tall. Funksjonen `twoGreatest(innliste)` skal skrive ut en liste bestående av *de to største elementene* i listen `innliste`, i Python-skallet, *i stigende rekkefølge*. Dersom listen har færre enn tre elementer, skal funksjonen skrive ut alle elementene. Funksjonen skal deretter *returnere* den *opprinnelige* listen `innliste`.

Når du bruker `lesTallListe()` og `twoGreatest()` sammen, skal du få et resultat som vist i dette eksempelet:

```
>>> liste = lesTallListe("")
Gi en liste av tall adskilt med blanke: 6 8 13 12 43 27 5 28 1
>>> print("liste før: ", liste)
liste før: [6, 8, 13, 12, 43, 27, 5, 28, 1]
>>> print("liste etter: ",twoGreatest(liste))
De to største elementene i listen er: [28, 43]
liste etter: [6, 8, 13, 12, 43, 27, 5, 28, 1]
```

Oppgave 3 (60%) Filen `us-temperatur.csv` er en kommaseparert tekstfil, det vil si at den inneholder linjer med data der hver linje inneholder datafelter separert med skilletegn. I dette tilfellet er skilletegnet semikolon, ‘;’. De første linjene i filen ser slik ut:

```
Contiguous U.S. Average Temperature Anomaly (degrees F);
200501;1,75
200502;2,5
200503;-0,88
```

Den første linjen er en *beskrivelse* av dataene, som vi ikke skal legge så mye vekt på i denne omgang. Hver av de resterende linjene representerer gjennomsnittlig temperatur-anomali for USA (målt i Fahrenheit), det vil si avvik fra en referanseverdi, for en gitt måned og år. For eksempel representerer tallet `200501` *året 2005 og måneden januar*. Den måneden ble det registrert en gjennomsnittstemperatur på 1,75 grader Fahrenheit over referanseverdien. I februar 2005 ble det registrert 2,5 F over referanseverdien, og så videre.

- a) (5%) Skriv en Python-funksjon `FtilC(F)` som tar en tallverdi F som innparameter. F representerer en temperaturanomali-verdi i Fahrenheit-grader. Funksjonen skal *returnere* verdien $F*5/9$, som er den tilsvarende (anomali)verdien i Celsiusgrader.

- b) (15%) Skriv en Python-funksjon `lesFil()` som leser filen `us-temperatur.csv`, ignorerer første linje, og returnerer en *liste av lister* av resten av dataene fra filen, på denne måten:

```
>>> lesFil()
[[2005, 1, 0.97], [2005, 2, 1.39], ...og så videre. ]
```

Hver av de «indre» listene skal altså være på formen

[år, måned som heltall, temperaturanomali *i Celsius*]

- c) (10%) Skriv en Python-funksjon `tempvariasjon(innliste)` som tar en liste, `innliste`, som innparameter. `innliste` skal være på det formatet som blir returnert av funksjonen `lesFil()` i b). Funksjonen skal returnere en tallverdi som er *maksimum over hele listen* av *det siste elementet i hver indre liste*, det vil si *maksimum temperaturanomali i Celsius* i hele `innliste`, avrundet opp til nærmeste heltallsverdi, slik at for eksempel

```
>>> tempvariasjon(lesFil())
```

skal returnere verdien 5

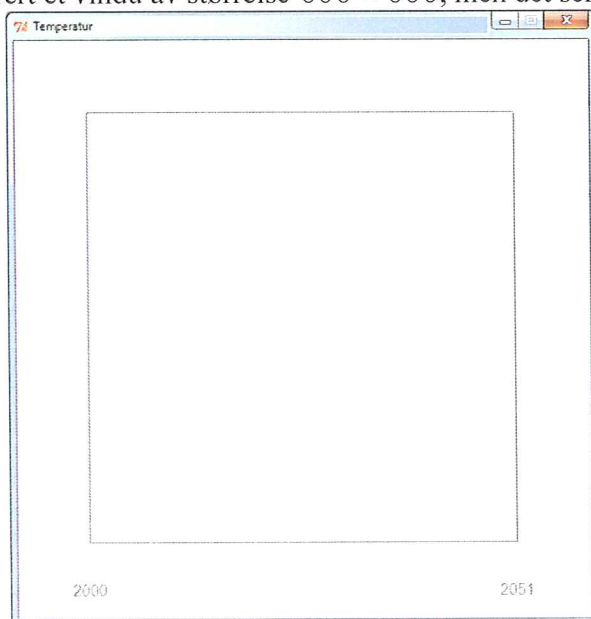
- d) (10%) I filen `EksamenINF109-H2014.py` finner du en metode

```
lagTegneVindu(fraaar, tilaar, temperaturvariasjon, size)
```

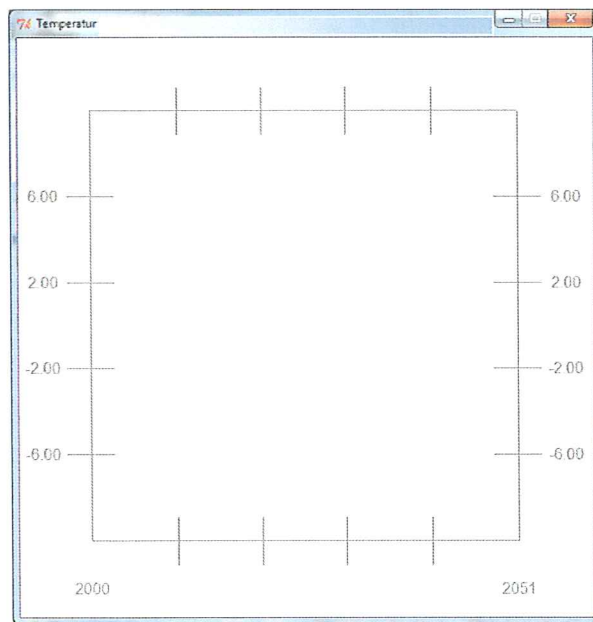
som lager, og returnerer, et tegnevindu av størrelse `size × size` med et aksesystem der aksene går fra `fraaar` til `tilaar + 1` (horisontalt) og fra `- temperaturvariasjon` til `+ temperaturvariasjon` (vertikalt), og med markeringer og forklaring langs aksene. *Uheldigvis er det en feil i metoden.* Når vi kaller metoden slik i Pythonskallet:

```
>>> lagTegneVindu(2000, 2050, 10, 600)
```

får vi riktignok generert et vindu av størrelse `600 × 600`, men det ser slik ut:



mens vi egentlig ønsker at vinduet skal se slik ut:



Finn og rett opp feilen i `lagTegneVindu()`. Hint: programkoden for å tegne streker og tekst langs aksene er på plass allerede, men det er en feil i de to innerløkkene.

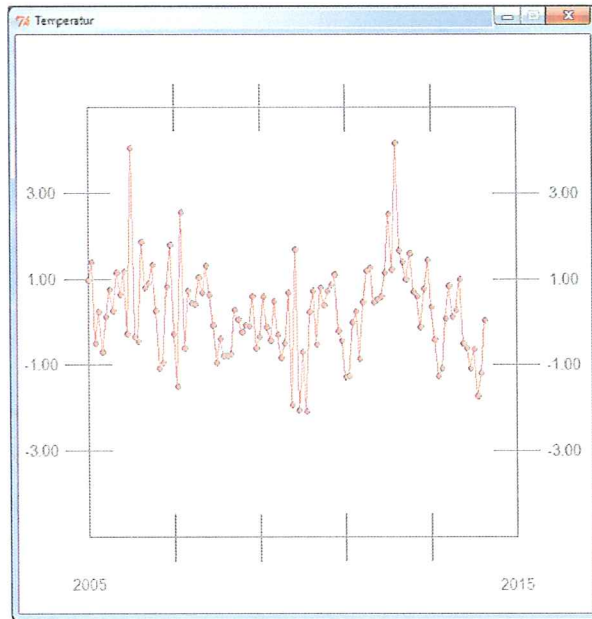
- e) (20%) Lag en Python-metode `tegne(fraaar, tilaar, maaned, liste)`. Parametrene er
- `fraaar`: en heltallsverdi som representerer årstall
 - `tilaar`: en heltallsverdi som representerer årstall
 - `maaned`: en liste av heltall. Hvert heltall forutsettes å være minst 1 og høyst 12.
 - `liste`: En liste av temperaturdata på det formatet som blir returnert av `lesFil()`.

Metoden `tegne()` skal

- finne største temperaturvariasjon ved å bruke `tempvariasjon()`-funksjonen,
- bruke `lagTegneVindu()` til å lage et tegnevindu,
- tegne alle data i `liste` som har årsverdier større eller lik `fraaar` og mindre eller lik `tilaar`, og månedsverdier som ligger i listen `maaned`, i tegnevinduet som ble returnert av `lagTegneVindu()`. Dataene skal tegnes med en liten fylt rød sirkel for hvert datapunkt, og en rød strek mellom datapunkt.

Når vi kjører metoden `temperatur()`, som ligger i filen `EksamenINF109-H2014.py`, skal vi for eksempel få følgende resultat:

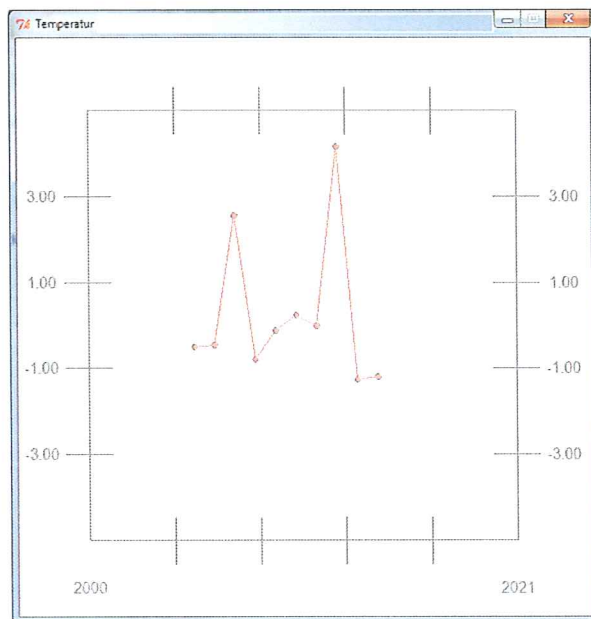
```
>>> temperatur()
Fra år:2005
Til år:2014
Gi en liste av måneder som tall adskilt med blanke:1 2 3 4 5 6 7 8 9 10 11 12
```



Fra år:2000

Til år:2020

Gi en liste av måneder som tall adskilt med blanke:3



Fra år:0

Lykke til!

Øyvind Ytrehus