



Alle teorispørsmål skal besvares og begrunnes. Alle oppgavene skal demonstreres til en studentassistent på sal. I oppgaver der du skriver programkode skal også denne vises fram. Lykke til!

1 Teori

- Hva er den binære ASCII-verdien av bokstaven A (stor a)? Hva er den i titallssystemet?
- Bruk ASCII tabellen til å oversette følgende til bokstaver:
01001100 01101001 01110011 01110000
- Hva er metadata?

2 Den store spørreundersøkelsen

I denne oppgaven skal du implementere et lite utdrag av en spørreundersøkelse om sosiale medier. Undersøkelsen er beregnet for kvinner og menn i aldersgruppen 16-25 år.

Spørreundersøkelsen implementeres ved hjelp av å skrive ut spørsmål og lese input fra brukeren i en while-løkke som kjøres så lenge brukeren ønsker å skrive inn data. Når alle spørsmålene er besvart, gjentas spørsmålene for en ny person.

Hold styr på fem **globale** tellere (variabler) under kjøringen av while-løkken: **antall_kvinner**, **antall_menn**, **antall_sosmedier**, **antall_facebook**, og **antall_timer_sosmedier**.

Hvis brukeren til enhver tid svarer **hade** på et spørsmål skal while-løkken avsluttes og verdiene av de fem tellerne skrives ut.

Deloppgave a til e skal altså implementeres inni while-løkken.

- Be brukeren skrive inn kjønn og alder. Med for eksempel **k** for kvinne og **m** for mann og lagre disse i variablene **kjonn** og **alder**.
- Sjekk om brukeren er innenfor tiltenkt aldersgruppe. Dersom brukeren er utenfor aldersgruppen, skriv en melding om at vedkommende ikke kan ta spørreundersøkelsen, og hopp tilbake til første spørsmål (kjønn).
- Dersom brukeren er innenfor tiltenkt aldersgruppe, spør om brukeren benytter seg av ett eller flere sosiale medier, med svaralternativene **ja** og **nei**. Lagre svaret som en variabel **aktiv_sosmedier**. Ta utgangspunkt i denne variabelen når du avgjør om brukeren trenger å svare på de neste to spørsmålene eller ikke.
- Du skal nå spørre om brukeren er medlem hos Facebook og lagre svaret som en variabel **medlem_facebook**, men spørsmålsteksten skal variere utifra om brukeren er kvinne eller mann.
 - Dersom brukeren er kvinne skal spørsmålsteksten være: *Mellom 55-60% av Facebook sine brukere er kvinner. Er du en av disse?*
 - Dersom brukeren er mann skal spørsmålsteksten være: *Mellom 40-45% av Facebook sine brukere er menn. Er du en av disse?*

- e) Spør brukeren hvor mange timer han/hun bruker i snitt daglig på sosiale medier og lagre svaret i variabelen `timer_sosmedier`
- f) Endre programmet ditt slik at de globale tellerne nevnt tidligere i oppgaven blir inkrementert i deloppgavene a til e.
- g) Kjør programmet ditt, utfør noen undersøkelser, og avslutt ved å skrive **hade**. Sjekk at statistikken (verdiene av de globale tellerne) som skrives ut på slutten er korrekt.
- h) Vil det være mulig å hente ut svarene fra spørreundersøkelsen etter at vi har avsluttet programmet? Hvor er de i så fall lagret? Eventuelt hvorfor ikke?

3 Forkorting av brøker

Et vanlig hverdagsproblem i matematikken er å finne den største divisoren av to tall, a og b . Eksempelvis vil det største tallet som deler både 30 og 20 være 10. En algoritme som løser dette problemet er Euklids algoritme.

Pseudokode for Euklids algoritme er som følger¹.

Eksempel 1: Euklids algoritme

- Gitt inputparametere a og b , begge heltall.
 - Gjenta så lenge b ikke er 0:
 - Tilegn den nåværende verdien av b til en ny variabel `gammel_b`.
 - Tilegn til b resten etter a dividert på b .
 - Tilegn til a verdien av `gammel_b`.
 - Returner a som resultat av algoritmen.
-

- a) Lag en funksjon `gcd` som tar inn to tall², utfører Euklids algoritme, og returnerer den største felles divisoren.
- b) Lag en funksjon `reduce_fraction` som tar inn to positive heltall fra brukeren, teller a og nevner b , bruker `gcd`-funksjonen definert i forrige deloppgave til å forkorte brøken $\frac{a}{b}$, og returnerer a og b .

For å returnere både a og b i `reduce_fraction` separer man de to returverdiene med komma: `return a, b`. For kalleren av funksjonen er det ganske likt. Man tilegner de to returverdiene av funksjonen til to variabler isteden for én: `a, b = funksjon_med_to_returverdier()`. Kall funksjonen for verdiene gitt i det neste eksempelet og skriv resultatet til skjermen på formen " a/b ".

Eksempel 2

For $a = 5$, $b = 10$ skriver programmet ut " $1/2$ "
For $a = 4$, $b = 2$ skriver programmet ut " $2/1$ ".
For $a = 42$, $b = 13$ skriver programmet ut " $42/13$ ".

¹Hvis du vil vite hvordan eller hvorfor denne algoritmen fungerer, huk tak i nærmeste tallteoribok.

²`gcd` \equiv greatest common divisor

4 Arbeidsdager

Et vanlig år har 52 hele uker og én dag, til sammen 365 dager. Det vil si at hvis året starter på en mandag, starter neste år på en tirsdag. Skuddår har en dag ekstra. Dvs. hvis et skuddår starter på en mandag, starter neste år på en onsdag. 1. januar 1900 var en mandag.

Skuddår er litt finurlig definert. Bruk følgende funksjon til å determinere om et år er et skuddår:

Kodesnutt 1

```
def is_leap_year(year):  
    if year % 400 == 0:  
        return True  
    elif year % 100 == 0:  
        return False  
    elif year % 4 == 0:  
        return True  
  
    return False
```

- a) Lag en funksjonen `weekday_newyear` som tar inn et årstall, og returnerer hvilken ukedag året starter på. Ukedager representeres med heltall, dvs. mandag = 0, tirsdag = 1, ..., søndag = 6.

Deloppgaven skal løses uten å bruke Pythons innebygde datofunksjoner.

Skriv ut årstall med tilhørende første ukedag i tekstlig format for år 1900 til og med 1919.

Eksempel 3

```
1900 man  
1901 tir  
1902 ons  
1903 tor  
1904 fre  
1905 son  
1906 man  
1907 tir  
1908 ons  
1909 fre  
1910 lor  
1911 son  
1912 man  
1913 ons  
1914 tor  
1915 fre  
1916 lor  
1917 man  
1918 tir  
1919 ons
```

- b) Lag funksjonen `is_workday(weekday)` som tar inn en ukedag, og returnerer `True` om uke-
dagen er arbeidsdag, og `False` ellers.
- c) Lag funksjonen `workdays_in_year(year)` som tar inn et årstall, og skriver ut antall ar-
beidsdager i det gitte året. Husk at skuddår har et en ekstra dag i februar.

Skriv ut antall arbeidsdager for årene 1900 til og med 1919.

Eksempel 4

1900	har	261	arbeidsdager
1901	har	261	arbeidsdager
1902	har	261	arbeidsdager
1903	har	261	arbeidsdager
1904	har	261	arbeidsdager
1905	har	260	arbeidsdager
1906	har	261	arbeidsdager
1907	har	261	arbeidsdager
1908	har	262	arbeidsdager
1909	har	261	arbeidsdager
1910	har	260	arbeidsdager
1911	har	260	arbeidsdager
1912	har	262	arbeidsdager
1913	har	261	arbeidsdager
1914	har	261	arbeidsdager
1915	har	261	arbeidsdager
1916	har	260	arbeidsdager
1917	har	261	arbeidsdager
1918	har	261	arbeidsdager
1919	har	261	arbeidsdager

5 Fibonacci

Fibonaccitallene er definert som følger.

$$f_n = \begin{cases} f_{n-1} + f_{n-2} & \text{hvis } n > 1 \\ 1 & \text{hvis } n = 1 \\ 0 & \text{hvis } n = 0 \end{cases}$$

Med $f_{n-1} + f_{n-2}$ menes summen av de to forrige fibonaccitallene.

[obligatorisk] a) Lag en funksjon som regner ut og returnerer det n -te fibonaccitallet f_n ved hjelp av iterasjon (dvs. ved å bruke en løkke).

[frivillig] b) Lag en funksjon som regner ut og returnerer det n -te fibonaccitallet f_n ved hjelp av rekursjon.

[frivillig] c) Lag en funksjon som returnerer en liste med alle fibonaccitallene opp til og med f_n .