



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

- a) Hva er pseudokode, og når får vi bruk for det? Skriv en enkel pseudokode for steking av speilegg.
- b) Hva er et flytdiagram, og hva skiller det fra pseudokode?
- c) Hva er debugging? Kan du gi noen eksempler på metoder man kan benytte når man debugger?
- d) Beskriv prinsippet “objektorientert programmering”. Er Python objektorientert?

2 Enkle betingelser

For å kontrollere programflyt benytter vi oss ofte av betingelser og boolske variabler. I forbindelse med å finne feil i kode (“debugge”), er det mange som i enkle tilfeller skriver ut verdien av variabler til skjerm som man ellers ikke ville skrevet ut, slik at man enklere kan forstå hvorfor programmet ikke gjør akkurat det man ønsker.

I denne oppgaven skal dere opprette en global boolsk variabel *debug*. Deretter skal dere definere en funksjon *add*, som tar inn to heltall, og skriver ut summen av disse til skjerm. Dersom *debug* er sann skal dere inne i funksjonen også printe verdien av tallene som summeres.

Eksempel på utskrift dersom *debug* er sann og tallene som sendes inn til funksjonen er 7 og 4:

Eksempel 1

```
Tallene som summeres er 7 og 4.  
11
```

3 En enkel løkkeoppgave

Skriv et program som leser heltallet n inn fra brukeren, og legger sammen tallserien $1^2 + 2^2 + 3^2 + \dots$ slik at summen blir størst mulig uten å overstige n . Hold styr på hvor mange elementer fra tallserien som er brukt i summen og skriv dette ut sammen med resultatet.

4 Strenger

Du har blitt gitt en liste med fornavn, `fornavn`, og etternavn, `etternavn`. Listene er like store, og er sortert slik at element nummer n i `fornavn` inneholder fornavnet til person n , mens element n i `etternavn` inneholder etternavnet til samme person.

Benytt (kopier) følgende kode som definerer de to listene.

Kodesnutt 1

```
fornavn = [
    'johan', 'eli', 'mats', 'lene', 'simon',
    'inger', 'henrik', 'kari', 'per']

etternavn = [
    'Hag', 'Hag', 'Basmestad', 'Grimlavaag', 'Kleivesund',
    'Fintenes', 'Svalesand', 'Molteby', 'Hegesen']
```

- Iterer forlengs gjennom de to listene, og skriv ut det fulle navnet på hver person.
- Denne gangen gjør du nesten det samme, men itererer baklengs i gjennom listen `etternavn`, og forlengs i gjennom `fornavn`.
Er noen av navnene like i forhold til forrige deloppgave? Hvorfor / hvorfor ikke?
- Du har sikkert merket at alle fornavnene mangler stor forbokstav. Endre programmet ditt til å skrive ut fornavnet med stor forbokstav for begge de forrige deloppgavene.

- 5** La oss si at du ønsker å poste (tweete) alle dine beste YouTube-videoer på Twitter slik at alle følgerne dine kan se hvor god smak du har. Dessverre så har Twitter en maksimumsgrense på hvor lange meldinger man kan skrive. Å lage flere poster er heller ikke et alternativ, for da kan det jo hende at ikke alle “retweeter” alle postene dine.

Du kommer dermed frem til den svært gode idéen med å bruke YouTube sin, korterte, alternative adresse, `youtu.be/<video-id>`, for alle linkene. En YouTube link i lang form, "`http://www.youtube.com/watch?v=<id>`" kan skrives i kort form som "`youtu.be/<id>`".

Gitt følgende liste av dine favorittvideoer.

Kodesnutt 2

```
videoer = [
    'http://www.youtube.com/watch?v=-840keiiFDE&t=1m40s',
    'http://www.youtube.com/watch?v=GpNSip5gyKo',
    'http://www.youtube.com/watch?v=sXX5drqRD9s',
    'http://www.youtube.com/watch?v=ZFngtBIxRPk',
    'http://www.youtube.com/watch?v=OZBWfyYtYQY',
    'http://www.youtube.com/watch?v=7LKHpm1UeDA']
```

Skriv kode som lager en liste med alle linkene i kortform.

6 Fibonacci

Fibonacci-tallene er definert som følger.

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

- a) Lag en program som regner ut og returnerer det n -te fibonaccitallet f_n ved hjelp av iterasjon.
- b) Kopier og modifier det nye programmet til å istedet returnere en liste med alle fibonaccitallene opp til og med f_n .