# INF1820, V2014 – Obligatorisk oppgave 1b
# Regulære uttrykk og endelige tilstandsmaskiner

## Innlevering: 14/2

Registrer svarene dine i en fil som angir brukernavnet ditt slik:
`oblig1b_brukernavn.py`
Innlevering av oppgaven skjer i Devilry. Se emnesiden for mer informasjon om reglement rundt innlevering, samt bruk av Devilry.
En perfekt løsning av denne oppgaven er verdt 100 poeng.

## 1 Enstavelsesord i "Call of the Wild" (30 poeng)

Her skal du analysere nok en tekst fra Project Gutenberg. Denne gangen skal vi jobbe med en hel roman, nemlig *Call of the Wild* av Jack London. Denne er fritt tilgjengelig fra `http://www.gutenberg.org/wiki/Main_Page`. Last ned teksten og pass på at du tar versjonen som er ren tekst, ikke HTML-versjonen.
Oppgaven består i å finne ut hvor mange av ordene i *Call of the Wild* som er enstavelsesord. I denne oppgaven kan du anta at et enstavelsesord består av null eller flere konsonanter, etterfulgt av en eller flere vokaler, etterfulgt av null eller flere konsonanter (d.v.s. at det kun er én gruppe vokaler i ordet). Du skal ikke skille mellom ord med stor og liten forbokstav. Dette kan du enten håndtere i det regulære uttrykket, eller ved å gjøre om alle ord til liten forbokstav.

(a) Skriv et Python-program som leser inn *Call of the Wild* fra en fil og bruker et regulært uttrykk for å telle antall enstavelsesord i teksten. Skriv ut antall ord, ikke hele listen med ord!

(b) Beskrivelsen av enstavelsesord gitt i oppgaven over er langt fra perfekt. Det finnes enstavelsesord som ikke dekkes av regelen, og det finnes ord som ikke er enstavelsesord som rapporteres som enstavelsesord.

Gi to eksempler på dette: ett ord som ikke er et enstavelsesord men som dekkes av regelen, samt ett ord som er et enstavelsesord men som ikke dekkes av regelen. Disse kan være hentet fra *Call of the Wild*, men trenger ikke være det.

## 2 Regulære uttrykk for datouttrykk (40 poeng)

Det finnes mange måter å angi dato'er på. For eksempel: "neste torsdag", "mandag 13 september, 2010", "02/07/2011".
En ting man kan gjøre med et system som henter ut datouttrykk er å lage en applikasjon som identifiserer møtetidspunkter i e-postene dine og automatisk legger disse til i kalenderen. (Merk dog at en slik applikasjon vil måtte være svært pålitelig for å være nyttig.)
Skriv regulære uttrykk som dekker følgende typer datouttrykk

- `neste/forrige ukedag`
- `ukedag dag måned, år`

der `ukedag` er en av mandag, tirsdag, ..., søndag, `måned` er en av januar, februar, ..., desember, og `dag` er et tall mellom 1 og 31. (Du trenger ikke sjekke at du ikke matcher 31 februar og andre umulige datoer). `år` er et tall mellom 1900 og 2099. Test de regulære utrykkene på minst 6 setninger som inneholder datouttrykk av hver type over og skriv ut resultatet.

## 3 Endelig tilstandsmaskiner (30 poeng)

I denne oppgaven skal vi bruke ekstern og fritt tilgjengelig programvare for å tegne FSA'er, nemlig JFLAP. Dette må dere laste ned selv fra `http://www.cs.duke.edu/csed/jflap/jflaptmp/`

- Last ned den nyeste versjonen som en .jar-fil. Du kan f.eks. bruke `wget`-kommandoen for å hente ned filen fra et terminalvindu slik:

  ```
  wget <URL>
  ```

- Kjør programmet fra kommandolinjen slik:

```
java -jar JFLAP.jar
```

Eksperimenter litt med programmet slik at du forstår hvordan det fungerer og håndterer følgende:

- hvordan legger man til tilstander?
- hvordan markerer man en tilstand som start-og sluttilstand?
- hvordan legger man inn kanter fra en tilstand til en annen?
- hvordan tester man maskinen på input?

Dersom du ikke får til dette, bør du se på en tutorial som ligger på JFLAP's nettside (`http://www.cs.duke.edu/csed/jflap/tutorial/`).

Tegn en endelig tilstandsmaskin i JFLAP som gjenkjenner følgende språk, der alfabetet er $\{a, b\}$. Alle tilstandsmaskinene skal være deterministiske.

(a) $\{ w \mid w$ inneholder minst tre $b$'er $\}$ (f.eks. skal *abbab* og *ababaababbb* aksepteres, men ikke *aaabaaa* og *ab*).

(b) $\{ w \mid$ hver oddetallsposisjon i $w$ er en $b \}$ (Maskinen kan godt akseptere den tomme strengen).

(c) $\{ w \mid w$ inneholder ikke substrengen *bba* $\}$ (Maskinen kan godt akseptere den tomme strengen).

Lagre de ferdige maskinene som .gif-filer (File → Save Image As), og lever dem sammen med koden din, som separate filer med navnene `3a.gif`, `3b.gif` og `3c.gif`.