# Uke 4, time 2

Fredag 12. september

# Oppgaver for løkker og samlinger

**Oppgave 1**

Lag et program som tar inn en setning som input fra en bruker. Som resultat skal programmet ta hvert ord og snu det baklengs, og skrive ut setningen med disse ordene, men med samme rekkefølge.

**Oppgave 2**

For å kunne konvertere poeng til en karakter, så kan man bruke en tabell, f.eks. slik A = 100-90, B = 89-80, C = 79-60, D = 59-50, E = 49-40, F = 39-0. Lag et program som tar en poengsum, og skriver ut en karakter. På slutten så skal den skrive ut hvor mange som fikk hver karakter.

**Oppgave 3**

En enkel måte å komprimere data er å telle antall forekomster av noe og deretter lagre det som verdi:antall. Som en begynnelse så lag et program som tar en rekke med tall, f.eks. i en streng som dette "0000113333333325222", skriv så ut i rekkefølge tallet og så antall forekomster. I eksempelet over, så kan svaret se ut noe som slikt: 0:4 1:2 3:7 2:1 5:1 2:3

**Oppgave 4**

En av de enkleste måtene å kryptere tekst, er det vi kaller for cæsarchiffer. Det fungerer ved å ta en bokstav og så hoppe et bestemt antall hakk opp eller ned i alfabetet. Så hvis vi bruker tallet 5, så vil "HEI" bli til "MJN". Hvis vi når enden av alfabetet så hoppe vi til begynnelsen, så Z blir til B i norsk alfabet. Hint er å lage alfabetet som en string og at den også fungerer som en liste med indekser

```
alfabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZÆØÅ"
```

Lag et program som tar et tall og en setning som input fra bruker og krypterer.

**Oppgave 5**

Lag et program som tar en lengre setning som input, og lager en ordbok som tar et ord som nøkkel, og som verdi det ordet som kommer etter. Så i setningen "jeg ser et hus og et tre", så blir resultatet

```
{ "jeg":"ser", "ser":"et", "et":"hus",
"hus":"og", "og":"et", "et":"tre" }
```

# Oppgaver

**Oppgave 1**

Lag et program som teller og skriver ut antallet ulike bokstaver som finnes i et ord. For eksempel har ordet “heisann” seks ulike bokstaver og “hei” tre ulike bokstaver.

Programmet skal fungere for ethvert ord. Test derfor programmet med ulike ord. Diskuter hva dere trenger i programmet og hvordan dere vil gå frem for å lage det før dere programmerer.

**Oppgave 2**

Dere skal fordele gjester langs den ene siden av et langt bord. På forhånd har dere delt gjestene inn i to grupper - introverte og ekstroverte - og ønsker å fordele disse på annenhver plass langs bordet. Det er alltid like mange introverte og ekstroverte gjester, men totalt antall gjester er vilkårlig (kan være hva som helst). Den endelige bordplasseringen med annenhver introverte og ekstroverte gjest er en liste.

Programmet skal avslutte med å skrive ut en ryddig oversikt som viser alle par av introverte og ekstroverte gjester. Husk at en gjest kan inngå i flere par; hvis for eksempel Per som er introvert er plassert mellom ekstroverte Lisa og ekstroverte Sigrid, inngår Per i to par.

Siden totalt antall gjester er vilkårlig (kan være hva som helst), test programmet med forskjellig antall gjester. Husk at antall introverte gjester alltid er likt antall ekstroverte. Diskuter hva dere trenger i programmet og hvordan dere vil gå frem for å lage det før dere programmerer.

**Oppgave 3**

En butikk har en liste med vareID-er (strenger). Lista inneholder alltid flere duplikater (like verdier) for én eller flere varenummer. Dere er ansatt for å hjelpe butikken med varetelling.

I den forbindelse ønsker butikken seg et program som tar vareID-er fra lista og lager en samling. Samlingen skal vise antallet av hver vare som erstatning for lista med duplikater. De ønsker dessuten en alfabetisk oversikt over vareID-er og antall fra samlingen skrevet ut i terminalen. Hint: dere kan bruke operatorene >, <, <=, >= og == for å sammenligne strenger.

Programmet skal fungere for enhver liste med vareID-er (strenger) med flere duplikater. Test derfor programmet med ulike lister. Diskuter hva dere trenger i programmet og hvordan dere vil gå frem for å lage det før dere programmerer.

**Oppgave 4: Ekstra utfordring**

Lag et program som fjerner alle nuller (hvis det finnes noen) i starten og slutten av en liste. Nuller som både har andre tall foran og bak seg i lista, skal ikke fjernes. Til slutt skal en liste som oppfyller de nevnte kravene skrives ut.

Eksempler

For listene [0, 0, 0, 0, 0] og [0] skal [] (tom liste) skrives ut. For [5] skal programmet skrive ut [5], og for listene [0, 0, 1, 2, 3] og [1, 2, 3, 0, 0] skal [1, 2, 3] skrives ut. For [1, 0, 2, 0, 3] skal programmet skrive ut [1, 0, 2, 0, 3] og for [1, 2, 3, 4] skal [1, 2, 3, 4] skrives ut. For [0, 0, 1, 2, 0, 3, 0, 0, 4, 0] skal programmet skrive ut [1, 2, 0, 3, 0, 0,4].

Programmet skal fungere for enhver liste. Test derfor programmet med listene nevnt ovenfor, i tillegg til andre lister. Diskuter hva dere trenger i programmet og hvordan dere vil gå frem for å lage det før dere programmerer.