Oppgave 3: Fotball-resultater + Oppgave 5

3a)

Skriv en funksjon `vinnerlag(hjemmelag, bortelag, hjemmemål, bortemål)` som tar som argument navn og antall mål for hjemme- og bortelag, og returnerer navnet på laget som vant (høyest antall mål). Om de to lagene har likt antall mål skal funksjonen returnere strengen "uavgjort". Du kan gå ut fra at argumentene til hjemmelag og bortelag er strenger

(type str), og at argumentene til hjemmemaal og bortemaal er heltall (type int). For eksempel skal kallet

```
vinnerlag("Brann", "Molde", 2, 3)
```

returnere

```
"Molde"
```

mens kallet

```
vinnerlag("Brann", "Molde", 2, 2)
```

skal returnere

```
"uavgjort"
```

3b)

Skriv en funksjon `forkort_lagliste(lagliste)` som tar som argument en liste av strenger som er lagnavn og returnerer en ny liste hvor ingen lagnavn finnes flere ganger i lista. Med andre ord, dersom samme streng opptrer flere ganger i lista som funksjonen mottar som argument, skal denne bare opptre én gang i lista som blir returnert. For eksempel skal kallet

```
forkort_lagliste(["Brann", "Molde", "Brann"])
```

returnere en liste

```
["Brann", "Molde"]
```

3c)

Skriv en funksjon legg_inn_null_mål(lagliste) som tar som argument en liste av strenger (lagnavn) og returnerer en ordbok som har hver streng fra lista (hvert lagnavn) som nøkkelverdi, med heltallsverdien 0 som innholdsverdi. For eksempel skal kallet

```
legg_inn_null_mål(["Brann","Molde"])
```

returnere en ordbok

```
{"Brann" : 0, "Molde" : 0}
```

3d)

Skriv en funksjon `hent_lagliste(filnavn)` som tar et filnavn som argument og returnerer en liste med alle lagnavn den finner i filen.

Funksjonen skal lese gjennom en fil (gitt av filnavnet mottatt som argument) hvor hver linje består av et navn på hjemmelag, navn på bortelag, antall mål hjemmelag, antall mål bortelag (separert med mellomrom).

Funksjonen skal returnere en liste som inneholder alle lagnavn som finnes i filen (som enten hjemmelag eller bortelag i ovennevnte format). Det gjør ikke noe (er uten betydning) om hvert lag opptrer flere ganger i den returnerte lista. Rekkefølgen av lagene i lista som returneres er også uten betydning.

En fil som skal leses gjennom kan altså se ut for eksempel som følger:

```
brann molde 2 0
sarpsborg molde 1 1
```

Funksjonen skal for dette eksempelet returnere en liste som inneholder lagnavnene brann, molde og sarpsborg, hvor eventuelle repetisjoner er uproblematisk - altså blir både

```
["brann", "molde", "sarpsborg"]
```

og

```
["brann", "molde", "sarpsborg",
"molde"]
```

godtatt som returverdi.

3e)

Skriv en funksjon `regn_poengsum(filnavn)` som tar et filnavn som argument og returnerer en ordbok med antall poeng for hvert lag ut fra kampresultat funnet i filen. Filen har samme format som i oppgave 3d, dvs hvor hver linje består av et navn på hjemmelag, navn på bortelag, antall mål hjemmelag, antall mål bortelag (separert med mellomrom).

Funksjonen `regn_poengsum` skal fortrinnsvis initielt kalle på funksjonene `hent_lagliste`, `forkort_lagliste` og `legg_inn_null_mål` (fra hhv oppgave 3d, 3b og 3c) for å konstruere en ordbok med null poeng for alle lag oppført i filen. Deretter skal den gå gjennom alle kampresultat (linjer) i filen og gi 3 poeng for vinnende lag, 0 poeng for tapende lag, samt 1 poeng til hvert lag i tilfeller av uavgjort.

Man kan gjerne kalle på funksjonen vinnerlag fra oppgave 3a for å hjelpe med utregning av poeng, men dette tillegges ikke vekt.

3f)

Skriv en funksjon `gull(lagoversikt)` som tar som argument en ordbok med lagnavn og antall mål (strenger som nøkkelverdier og heltall som innholdsverdier) og returnerer navnet på laget med høyest poengsum (nøkkelverdien med høyeste tilhørende innholdsverdi). Du kan anta at ett lag har høyest poengsum (ikke to lag med samme poengsum), og at det finnes noen lag med positiv poengsum i ordboken mottatt som argument. For eksempel skal kallet

```
gull({"Brann" : 2, "Molde" : 3})
```

returnere

```
"Molde"
```

3g)

Skriv en prosedyre `finn_gull(filnavn)` som tar et filnavn som argument og skriver navnet på laget med flest poeng ut til terminalen. Filen har samme format som i oppgave 3d. Poeng regnes ut basert på vinnende og tapende lag for hver kamp som beskrevet i oppg 3e. Som i oppg 3f kan du anta at ett lag har høyest poengsum (ikke to lag med samme poengsum), og at det finnes minst ett lag med positiv poengsum. Du kan benytte deg av kall på hvilke funksjoner/prosedyrer du ønsker fra deloppgaver 3a-3f for å lage en enklest mulig løsning for prosedyren `finn_gull` (du kan kalle på funksjoner/prosedyrer fra andre deloppgaver ut fra den beskrevne funksjonaliteten, uavhengig av om du selv har løst disse deloppgavene eller ikke).

Oppgave 5

Skriv en funksjon `godkjenn(familier)` som tar som argument en liste av lister med aldre for personer i familier. Hver familie er representert med en liste av aldre for alle medlemmene i familien i vilkårlig rekkefølge.

En familie med tre medlemmer av alder 30, 10 og 2 år, kan dermed f.eks. være representert av en liste `[10,2,30]`. Alle aldre er heltall. Funksjonen godkjenn skal ta inn en liste av slike lister (altså flere familier). Om man i tillegg til nevnte familie har en annen familie med aldre 20 og 1, vil dette altså kunne være representert som `[ [10,2,30], [20,1] ]`.

Skriv funksjonen `godkjenn` slik at den sjekker at alle familier har minst én myndig person, altså at alle lister har minst en verdi som er større eller lik 18.

Så lenge alle familier har minst én myndig person, skal funksjonen returnere True. Dersom én eller flere familier mangler myndig person (at det for minst én av listene er slik at alle verdier er under 18) skal den returnere False. For eksempel skal kallet

```
godkjenn([ [10,2,30], [20,1] ])
```

returnere

```
True
```

mens kallet

```
godkjenn([ [10,2,30], [10,1] ])
```

skal returnere

```
False
```