

Den ultimate karakterfordeling

Læringsmål:

- If-setninger
- Tilfeldige tall

Pensum

- Kap 1.6
- Kap 2.1.3
- Kap 3.7
- Kap 4.1 - 4.3

En sprø professor har et frivillig fag om "tilfeldighet". Han bedømmer eksamen etter to faktorer: En tilfeldig poengsum og en svært uformell gruppering av håndskrift: **fin** eller **stygg**. Fin skrift trekker en hel karakter opp, mens stygg skrift trekker en hel karakter ned. Han gir ikke ut karakterer over A eller lavere enn F.

Det skal lages et program som skriver ut hvor mange poeng en elev fikk, karakteren dette tilsvarer, og hvorvidt håndskriften trakk eleven opp eller ned. En eksempelutskrift fra å kjøre programmet kan være som følger:

"Du fikk 87 poeng på eksamen, en B. Fordi du skrev stygt ble dette en C."

Vi skal gjøre dette ved å bryte ned problemet i mindre deler.

a) Lag en funksjon som returnerer en bokstavkarakter basert på antall poeng. Prototypen skal se slik ut:

```
function grade = num2grade(points)
```

Poenggrensene er som følger:

| A | B | C | D | E | F |
|--------|-------|-------|-------|-------|------|
| 100-89 | 88-77 | 76-65 | 64-53 | 52-41 | 40-0 |

Skriv ut en feilmelding om antall poeng ikke er et heltall eller er utenfor skalaen.

Tips: Bruk den innebygde funksjonen, `error()`, til å skrive ut feilmeldingen.

Eksempel på kjøring:

```
>> num2grade(65)
ans =
C
>> num2grade(110)
Error using num2grade (line 6)
points må ligge i intervallet [0, 100]!
```

b) Skriv funksjonen `evaluateGrade(points, handwriting)`. Den skal ha to parametre: en poengsum (heltall) og en boolsk verdi for kvaliteten på håndskriften. Fin håndskrift = **true**, stygg håndskrift = **false**. Den skal deretter returnere hva bokstavkarakterer blir når håndskriften tas hensyn til.

Tips: Bruk `num2grade` og det at `char('B' - 1) = 'A'`.

Eksempel på kjøring:

```
>> evaluateGrade(83, true)
ans =
A
>> evaluateGrade(94, true)
ans =
A
>> evaluateGrade(56, false)
ans =
E
>> evaluateGrade(34, false)
ans =
F
```

c) Det er behov for en funksjon `randomPoints()` som generer en tilfeldig poengsum (heltall!) fra 0 til 100, og returnerer denne. Lag denne og test at den fungerer med: `num2grade(randomPoints())`.

Eksempel på kjøring:

```
>> randomPoints()
ans =
    19
>> num2grade(randomPoints())
ans =
C
```

d) Lag også en funksjon `randomHandW()` som returnerer en tilfeldig håndskrift, **true** (fin) eller **false** (stygg).

Eksempel på kjøring:

```
>> randomHandW
ans =
    1
>> randomHandW
ans =
    0
>> randomHandW
ans =
    0
```

e) `num2grade(points)`, `evaluateGrade(points, handWriting)`, `randomHandW()` og `randomPoints()` skal settes sammen. Funksjonen `grades()` skal bruke funksjonene fra de tidligere deloppgavene til å generere en tilfeldig poengsum og håndskrift, og deretter skrive ut til brukeren hva poengsummen ble, hvilken karakter det tilsvarer, og hvordan håndskriften påvirket karakteren.

Eksempel på kjøring:

```
>> grades
Du fikk 58 poeng på eksamen, en D. Fordi du skrev fint ble dette en C
>> grades
Du fikk 96 poeng på eksamen, en A. Fordi du skrev fint ble dette en A
>> grades
Du fikk 93 poeng på eksamen, en A. Fordi du skrev fint ble dette en A
```