

Oppgave 3

Formål: Lage tester for de forskjellige typene i golang med utgangspunkt i pakken sum fra REP_ICA02

Lage tester som produserer «feil»

```
# command-line-arguments
.\sum_test.go:15: constant 128 overflows int8
.\sum_test.go:33: constant 2147483648 overflows int32
.\sum_test.go:50: constant -2 overflows uint32
.\sum_test.go:87: constant 9223372036854775809 overflows int64
FAIL    command-line-arguments [build failed]
```

Her kan vi se at vi har fått en feil på alle de forskjellige typene. Feilen som går igjen er «overflow», og betyr at vi har brukt tall som er utenfor «rangen» til de forskjellige datatypene. Oversikt over forskjellige typer og deres «range» kan finnes [her](#).

En ting vi kan notere oss er at det ikke dukket opp noen feil på Float64, dette er fordi disse verdiene er utrolig store, og det var faktisk litt vanskelig å få en overflow feil på Float64.

Prøv ut verdier som ikke passerte test-rutiner

Ved å endre slik at alle verdier blir konvertert til riktig format og kjøre de alle samtidig kan vi enkelt prøve ut de verdiene som ikke fungerte, for eksempel 127 + 1

```
For sum.go: 127
Resultat som type Int8
-128
Resultat som type Int64
128
Resultat som type Int32
128
Resultat som type Float64
128
Resultat som type Uint32
128
```

Her kan vi se at alle andre typer enn Int8 klarte å håndtere verdien, som forventet. Vi kan fortsette i evigheter, men så lenge vi bommer på «rangen» til en type, så vil vi få returnert et ugyldig resultat (se -128 over)

Foreslå en løsning som gjør pakken «sum» trygg for brukeren

For å gjøre denne løsningen tryggere så kan vi bruke en eventuell if setning, switch eller lignende for å konvertere til den numeriske typen som passer best. Vi kan gjøre dette ved å bruke et and eller or statement, slik at vi alltid vil ende opp med en passende type. Eks hvis tallet er mindre enn 128 og ikke er negativt kan vi bruke Uint8.