



UNIVERSITETET I BERGEN

REPETISJON

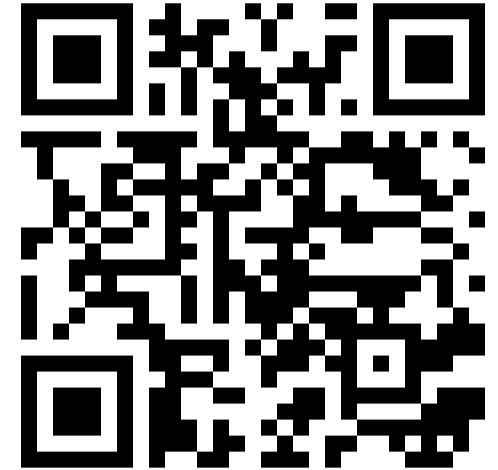
INF100

HØST 2022

Torstein Strømme

PRØVEEKSAMEN

- Mandag 14. november 14:15 – 18:15
- Egget i Studentsenteret
 - eller hjemmefra
- 100% frivillig
- Blir *ikke* rettet av oss
- Sensorveiledning blir publisert → bytt besvarelse med en venn
- Påmeldingsfrist 10. november 23:45:
<https://skjemaker.app.uib.no/view.php?id=13628096>



VANLIGE FEIL

- Å ikke prøve
- Manglende identifikasjon av delproblemer
 - For mange ting på samme linje
 - For mange ting i samme funksjon (manglende bruk av hjelpefunksjoner)
- Dårlige variabelnavn
 - `for i in a:` vs `for i in range(len(a)):`

VANLIGE FEIL

- Presedens
 - x and y in z
 - x == 3 or 4
- Funksjoner
 - print vs return
 - skop for variabler
 - return i destruktive funksjoner
- Lister
 - indeks vs. element
 - løkker over indeks vs elementer
 - modifisering av lister i en løkke
- Løkker
 - For tidlig return
 - Hva skal gjøres én gang, hva skal gjentas flere ganger?

EKSAMENSSTRATEGI

- Les gjennom alle oppgavene
 - Ikke kast bort tid på et problem du ikke har en klar plan for før du har lest nøye gjennom alle oppgavene.
- Første gang du leser problemet: identifiser delproblemer
 - Skriv ned kommentarer/idéer. Hva skal gjentas flere ganger? Hjelpesfunksjoner gjør hva?
 - Selv om du ikke klarer løse hele problemet, kjenner du kanskje igjen en del av det du klarer å løse.
- Prioriter oppgaver du kan godt og oppgaver med mye poeng.

EKSAMENSSTRATEGI

- Bruke selvbeskrivende variabelnavn
 - Lettere for deg selv å forstå hva du driver med
 - Lettere for sensor å forstå hva du driver med
- Bruk kursnotatene
- Søk på internett (og SITER KILDER)
- Spis godt før du starter
- Sov godt før du starter
- Gå en tur før du starter

PRESEDENS

- Operatøren med *lavest* presedens evalueres *sist*
- Dersom flere operatører har lik presedens, evalueres den til *høyre sist*
 - (Unntak: eksponentiering)
- Presendenstabell:
<https://inf100.i.uib.no/notat/presedens/>

PRESEDENS

www.menti.com

1237 0585



VANLIGE FEIL

```
def print_longest_words(w1, w2, w3):  
    longest = max(w1, w2, w3)  
    if len(w1) == longest:  
        return w1  
    elif len(w2) == longest:  
        return w2  
    elif len(w3) == longest:  
        return w3
```

*Feil: sammenligner
feil typer ting*

VANLIGE FEIL

```
def joker(x1, x2, x3):  
    if x1 <= 4:  
        print("opp")  
    else:  
        print("ned")  
        if x2 <= 4:  
            print("opp")  
        else:  
            print("ned")  
            if x3 <= 4:  
                print("opp")  
            else:  
                print("ned")
```

Feil: feil med innrykk gjør at kode får flere betingelser enn den skulle hatt.

VANLIGE FEIL

```
def is_even_positive_int(x):  
    return x==int and x >= 0 and x % 2 == 0
```

*Feil: sammenligner
feil typer ting*

*Tips: gjør én ting
om gangen*

VANLIGE FEIL

```
def are_all_ints(x1, x2, x3):  
    if type(x1) == int:  
        return True  
    else:  
        return False  
    if type(x2) == int:  
        return True  
    else:  
        return False  
    if type(x3) == int:  
        return True  
    else:  
        return False
```

```
def are_all_ints(x1, x2, x3):  
    for x in (x1, x2, x3):  
        if type(x) == int:  
            return True  
        else:  
            return False
```

*Feil: returnerer før man
er ferdig å regne ut svaret*

VANLIGE FEIL

```
def approx_area_under_g(x_lo, x_hi):  
    running_total = 0  
    for x in range(x_lo, x_hi):  
        running_total += g(x)  
    return running_total
```

*Feil: returnerer inne i en
løkke før man er ferdig
å regne ut svaret*

VANLIGE FEIL

```
def find_nth_occurrence(word, character, n):  
    found_so_far = 0  
    for i in word:  
        if word[i] == character:  
            found_so_far += 1  
            if found_so_far == n:  
                return i  
    return -1
```

*Feil: blander indekser
og elementer i en
løkke over lister*