

i Forside

UNIVERSITET I OSLO

Det matematisk-naturvitenskapelige fakultet

Midtveiseksamen IN-KJM1900 og MAT-IN1105

Dato: 7 oktober 2020

Vedlegg: Ingen

Tillatte hjelpemidler: Ingen

- Oppgavesettet består av 21 flervalgsoppgaver. Maksimal oppnåelig poengsum er 25.
- All kode i oppgaveteksten er skrevet i Python 3.
- På de fleste oppgavene er det ett riktig svaralternativ. Riktig svar gir 1 poeng, feil svar gir 0 poeng. Man får ikke minuspoeng for feil svar på disse oppgavene, så det lønner seg alltid å svare.
- Oppgave 17, 18, og 19 har ett eller flere riktige svaralternativer. Det gis 1 poeng for hver riktig avkrysning, -1 for feil avkrysning, 0 poeng for ubesvart. Minste poengsum på hver av oppgavene er 0. Det lønner seg alltid å markere minst ett svar.
- Oppgave 20 har seks svar. Det gis 0.33 poeng for hvert riktig svar og 0 poeng for feil eller ubesvart, maksimalt 2 poeng på oppgaven.
- Siden maksimal poengsum er 25 kan man regne ut at det må være til sammen seks korrekte svaralternativer for oppgavene 17, 18, og 19. Denne antagelsen er korrekt, men antallet korrekte alternativer på hver av de tre oppgavene vil ikke bli oppgitt.

1 Hva skrives ut?

Hva skrives ut i terminalvinduet når følgende kode kjøres?

```
a = 3
b = a
a = 4
print('b =', b)
```

Velg ett alternativ

- b = 3
- b = 4
- 3
- En feilmelding
- 4



Maks poeng: 1

2 Hva skrives ut?

Hvilket tall vil bli skrevet ut?

```
a = 0
if a == 0:
    a += 1
if a == 1:
    a += 1
else:
    a += 2
print(a)
```

Velg ett alternativ:

- 0
- 3
- 1
- 2



Maks poeng: 1

3 Hva skrives ut?

Hva skrives ut i terminalvinduet når følgende kode kjøres?

```
a = 3
b = 5
def prod(a,b):
    return a*b
s = prod(a,a)
a = 5

print(a,s)
```

Velg ett alternativ

- 5 9
- 9 5
- En feilmelding
- 3 9



Maks poeng: 1

4 Hvilket funksjonskall?

Den følgende funksjonen estimerer den deriverte til en funksjon f i et punkt x . Både f og x sendes som argumenter til funksjonen.

```
import numpy as np
def num_diff(f,x,h=1e-6):
    return (f(x+h)-f(x))/h
```

Vi ønsker å bruke funksjonen til å evaluere den deriverte til eksponentialfunksjonen i $x=1$. Hvilket funksjonskall er riktig?

Velg ett alternativ

- `d = num_diff(np.exp(),1)`
- `d = num_diff(np.exp,1,h=1e-8)`
- `d = num_diff(np.exp(x),1)`
- `d = num_diff(np.exp(1))`
- `d = num_diff(np.exp(x),1,1e-8)`



Maks poeng: 1

5 Hva skrives ut?

Hva skrives ut i terminalvinduet når følgende kode kjøres?

```
for i in range(1,4):
    for j in range(i-1):
        if j < i:
            print(j,i, end=' ')
```

(Argument `end=' '` som sendes til `print`-funksjonen erstatter det vanlige linjeskiftet med et mellomrom.)

Velg ett alternativ

- `0 0 0 1 0 2 1 2`
- `0 1 2 3`
- `2 0 3 0 3 1`
- `0 2 0 3 1 3`



Maks poeng: 1

6 Hva skrives ut?

Hva skrives ut i terminalvinduet når følgende kode kjøres?

```
dx = 0.5  
c = [dx*i for i in range(10)]  
print(c[-2])
```

Velg ett alternativ

- En feilmelding
- 4.5
- 4.0
- 4
- 3.5



Maks poeng: 1

7 Hva skrives ut?

Hva skrives ut når følgende kode kjøres?

```
def square(z):  
    return z**2  
  
def test_square():  
    z = 2.0  
    tol = 1e-10  
    expected = 4.0  
    computed = square(z)  
    success = abs(expected-computed) < tol  
    msg = 'Something wrong in square'  
    assert success, msg
```

test_square()

Velg ett alternativ

- Ingenting
- AssertionError: Something wrong in square
- Success



Maks poeng: 1

8 Hva skrives ut?

Hva skrives ut når følgende kode kjøres?

```
import sys
```

```
try:
```

```
    v0 = float(sys.argv[0])
```

```
    g = float(sys.argv[1])
```

```
    print(v0*t-0.5*g*t**2)
```

```
except IndexError:
```

```
    print('Please provide two command line arguments')
```

```
except ValueError:
```

```
    print('The arguments must be numbers')
```

Koden er i en fil med navn ball.py, og kjøres på følgende måte:

```
Terminal> python ball.py 5 9.81
```

Velg ett alternativ

- Please provide two command line arguments
- Ingenting
- The arguments must be numbers
- 1.2342



Maks poeng: 1

9 Hva skrives ut?

Hva skrives ut når følgende kode kjøres?

```
from numpy import *
```

```
a = linspace(2,6,5)
```

```
b = a + 2.0
```

```
print(b)
```

Velg ett alternativ

- En feilmelding
- [4. 5. 6. 7. 8.]
- [2. 3. 4. 5. 6.]
- [2. 3. 4. 5. 6. 2.]



Maks poeng: 1

10 Hva skrives ut?

Hva skrives ut når følgende kode kjøres?

```
x = 7  
y = 13  
a = 6  
b = 14
```

```
c = x > a or y > b  
print(c)
```

Velg ett alternativ

- False
- 12
- True
- En feilmelding
- 8



Maks poeng: 1

11 Hva skrives ut?

Hva skrives ut i terminalvinduet når følgende kode kjøres?

```
def length(a):  
    r = 0  
    for e in a:  
        r += 1  
    return r
```

```
b = list(range(1,10))  
print(len(b), length(b), len(b) == length(b))
```

Velg ett alternativ

- 10 10 True
- 10 10 0
- 9 9 True
- En feilmelding
- 9 9



Maks poeng: 1

12 Hvilket funksjonskall?

Python-funksjonen `plot_function` tar en funksjon `f` og to verdier `x0` og `x1` som argumenter, og plotter $f(x)$ i intervallet $[x_0, x_1]$.

```
import numpy as np
import matplotlib.pyplot as plt
```

```
def plot_function(f,x0,x1,n=100):
    x = np.linspace(x0,x1,n+1)
    plt.plot(x,f(x))
    plt.xlabel('x')
    plt.ylabel('f(x)')
    plt.show()
```

Vi ønsker å bruke funksjonen til å plote $f(x) = \sin(x)$, for x mellom 0 og 1. Hvilket funksjonskall er riktig?

Velg ett alternativ

- `plot_function(np.sin, 0)`
- `plot_function(np.sin, 1)`
- `plot_function(np.sin(),0,1)`
- `plot_function(np.sin,0,1)`
- `plot_function(np.sin(),[0,1],n=100)`
- `plot_function(np.sin, [0,1])`



Maks poeng: 1

13 **Hva skrives ut?**

Filen 'hurdles.txt' ser ut som følger:

2020 46.87 Karsten Warholm
2019 46.92 Karsten Warholm
2018 46.98 Abderrahman Samba
2019 46.98 Rai Benjamin
1992 46.78 Kevin Young
1983 47.02 Edwin Moses
2018 47.02 Rai Benjamin

Hva skrives ut av følgende kode?

```
infile = open('hurdles.txt','r')  
fastest = 60.0  
for line in infile:  
    t = float(line.split()[1])  
    if t < fastest:  
        fastest = t  
        r = line  
print(r)
```

Velg ett alternativ

- 2018 47.02 Rai Benjamin
- 1992
- 46.78
- Karsten Warholm
- 2020 46.87 Karsten Warholm
- 1992 46.78 Kevin Young



Maks poeng: 1

14 Hva skrives ut?

Hva skrives ut i dette programmet?

```
old_list = ["You", "don't", "understand!"]
new_list = []
for i in range(0, 3, 2):
    word = old_list[i]
    new_list.append(word)
print(new_list)
```

Velg ett alternativ:

- ['You', "don't", 'understand!']
- ['You']
- ['understand!']
- []
- ['You', 'understand!']



Maks poeng: 1

15 Hva skrives ut?

Hva skrives ut i dette programmet?

```
import numpy as np
N = 3
F = np.zeros(N)
for i in range(N):
    for j in range(i+1,N):
        F[i] = F[i-1] + 2

print(F)
```

Velg ett alternativ:

- [2. 4. 0.]
- [2. 4. 2.]
- [2. 4. 6.]
- [0. 2. 4. 6. 8.]



Maks poeng: 1

16 **Hva skrives ut?**

Hva skrives ut i dette programmet?

```
some_numbers = [1, 3, 4]
other_numbers = [3, 5, 5]
times = 0
while sum(some_numbers) < 24:
    if some_numbers[1] + some_numbers[2] > 6:
        some_numbers += other_numbers
    else:
        some_numbers += [1,2]
    times += 1

print(times)
```

Velg ett alternativ:

- 5
- 4
- 2
- 0
- 1
- 6
- 3



Maks poeng: 1

17 Feil i kode

Programmet nedenfor prøver å benytte følgende algoritme for å tilnærme kvadratroten til et tall a :

$$x_{n+1} = 0.5(x_n + \frac{a}{x_n})$$

```
1 from numpy import sqrt
2 a = 12
3 x0 = 2.5
4 n = 3
5 x = x0
6
7 for i in range(n):
8     x = 0.5*(x0 + a/x)
9
10 error = abs(x-np.sqrt(a))
11 print(f'The square root of {a} is approximately {x} with an error of {error}')
```

I hvilke(n) linje(r) er det feil i koden?

Velg ett eller flere alternativer

- linje 7
- linje 10
- linje 5
- linje 8
- linje 11



Maks poeng: 2

18 Sammenlikninger

Vi har følgende kode.

```
import numpy as np
a = [x + 2 for x in range(1,10,3)]
b = []
c = np.array([1,2,3])
x = 2
for i in range(1,3):
    b.append((1+2*i)**2)
while x < 3:
    c[x] = x
    x += 2
```

Hvilke(t) av følgende uttrykk har verdien True etter at koden har kjørt?

1 poeng gis for riktig markert alternativ, -1 for feil markert alternativ, og 0 poeng for umarkerte alternativer. Minimum total poengsum på oppgaven er 0 poeng.

Velg ett eller flere alternativer

- `len(c) != len(b)`
- `a[-1] == b[0]`
- `len(c) > len(a)`
- `a==b!=c`
- `a==b==c`



Maks poeng: 2

19 **Hvilken kode er riktig?**

Vi ønsker å beregne hvor lang tid det tar før vi fordobler et beløp vi har satt i banken, gitt at renta er lik 3 % i alle år. Vi starter med følgende variabler:

```
start_sum = 1000
money = start_sum
interest = 3 # percent
interest_rate = 1.03
time = 0
end_sum = 2*start_sum
```

Hvilke(n) kodesnutt(er) gjør det vi ønsker?

(A)

```
for i in range(100):
    if money < end_sum:
        money = money + money*interest_rate
        time += 1
```

(B)

```
for i in range(100):
    if money < end_sum:
        money = money*interest_rate
        time += 1
```

(C)

```
while time < 100:
    money = money*interest_rate
    time += 1
```

(D)

```
for time in range(100):
    money = money*interest_rate
    time += 1
```

(E)

```
while money < end_sum:
    money = money + money*interest/100
    time += 1
```

Velg ett eller flere alternativer

B



C

D

E



A

Maks poeng: 2

20 **Boolske uttrykk**

Indiker for hver rad om verdien av det logiske (boolske) uttrykket er True eller False, eller om det er feil formulert og vil gi en feilmelding. 1/3 (0.33) poeng blir gitt for hver riktige linje, maksimalt 2 poeng på oppgaven.

Finn de som passer sammen

	True	Gives error message	False
False != (False or True)	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>
True != False	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>
False != (4 >= 3)	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>
not (3 != 4)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> ✓
not (True != False)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> ✓
not False or True	<input type="radio"/> ✓	<input type="radio"/>	<input type="radio"/>

Maks poeng: 1.98

21 **Skriv om koden**

Følgende program bruker en løkke til å konvertere grader fahrenheit til grader celsius.

```
C = []  
F = 0  
while F < 40:  
    C.append((F-32)*5/9)  
    F += 5  
print(C)
```

Hvilken av kodene nedenfor gir samme output?

(A)
C = []
F = 0
for F in range(40):
 C.append((F-32)*5/9)
 F += 5
print(C)

(B)
C = []
F = 0
for F in range(0, 40, 5):
 C.append((F-32)*5/9)
 F += 5
print(C)

(C)
C = []
F = 0
for F in range(40, 5):
 C.append((F-32)*5/9)
print(C)

(D)
C = []
F = 0
for F in range(40, 5):
 C[i] = ((F-32)*5/9)
print(C)

Velg ett alternativ:

- A
- D
- C
- B



Maks poeng: 1

