Forside

UNIVERSITY OF OSLO
Maculty of Mathematics and Natural Sciences

Mid-term exam IN1900, MAT-IN1105, IN-KJM1900, IN-GEO1900
Date & time: October 11, 2023, 15:00-19:00
Attachments: None
Permitted aids: None. A calculator is available in Inspera

- The exam set consists of 22 multiple choice questions. The maximum score is 25 points.
- All the code in the exam is written in Python 3.
- Most of the questions have one correct answer. A correctly marked answer gives 1 point, wrong or missing answer gives 0 points. There are no negative points for wrong answers, so one should always mark an answer.
- Questions 9, 10, og 22 have 4 answers each. It is given 0.5 points for each correct answer and 0 points for wrong or unmarked answers, maximum 2 points total for each of these questions.

1 Hva skrives ut?

What is printed in the terminal window when the following code is run?

```
x = 4
y = x
x = x + y
print(x, y)
```

Select one alternative:

○ 8 4

○ 8 8

○ An error message

○ 4 4

Maximum marks: 1

2 Hva skrives ut?

What is printed in the terminal window when the following code is run?

```
l = []
for i in range(3):
   l.append(i)
print(l)
```

Select one alternative:

○ An error message

○ [1, 2, 3]

○ [0, 1, 2]

○ [0, 1, 2, 3]

Maximum marks: 1

3 Hva skrives ut?

What is printed in the terminal window when this code is run?

```
hello = "Hello students! Welcome to the midterm exam!"
words = hello.split()
print(type(words),type(words[-1]))
```

Select one alternative:

○ An error message

○ <class 'list'> <class 'str'>

○ <class 'str'> <class 'str'>

○ <class 'str'>

Maximum marks: 1

4 Hva skrives ut?

What is printed when this code is run?

```
import math
r = 1
t = 2.0
A = math.exp(r * t)
print(type(r), type(A))
```

Select one alternative:

○ <class 'int'> <class 'float'>

○ <class 'int'> <class 'int'>

○ <class 'float'> <class 'float'>

○ <class 'float'>

Maximum marks: 1

5 Hva skrives ut?

What is printed when this code is run?

```
x = 8
y = 6
z = 3
print(y > x or x > z)
```

Select one alternative:

○ False

○ An error message

○ True

○ False or True

Maximum marks: 1

6 Hva skrives ut?

What is printed when this code is run?

```
s = 0
for n in range(1, 9, 3):
    s += n
print(s)
```

Select one alternative:

○ 21

○ 12

○ 9

○ 18

Maximum marks: 1

7 Hva skrives ut?

What is printed when this code is run?

```
def line(t, a, b):
    return a * t + b

def square(x):
    return x**2

a = 2
b = 1
x = 3
t = 2
print(square(line(0, 1, 4)))
```

Select one alternative:

○ 9

○ An error message

○ 16

○ 25

Maximum marks: 1

8 Hva skrives ut?

What is printed when this code is run?

```
x = []
n = 100
dx = 1.0 / n

for i in range(n + 1):
    x.append(i * dx)

print(len(x), x[-1])
```

Select one alternative:

○ 101 1.0

○ 100 1.0

○ 100 0.99

○ 101 1.01

Maximum marks: 1

9 Lister og Numpy arrays

Assume that this code is run.

```
import numpy as np
dx = 0.01
x = [i * dx for i in range(101)]
y = np.linspace(0,1,101)
```

What are the values of the boolean expressions?

Please match the values:

	True	False
x[50] > 2 * y[50]	○	○
len(x) == 100	○	○
x[-1] <= 0.5	○	○
len(y) == len(x)	○	○

Maximum marks: 2

10 Løkker

Assume that this code is run.

```
year = 2002
years1 = []
years2 = []

while year <= 2023:
   years1.append(year)
   year += 1

for year in range(2002,2023):
   years2.append(year)
```

What are the values of the boolean expressions?

Please match the values:

	True	False
years1[1] == years2[1]	○	○
years1[0] == years2[0]	○	○
len(years1) == len(years2)	○	○
years1[-1] == years2[-1]	○	○

Maximum marks: 2

11 Hvilken funksjon?

Here are three alternative implementations of the function **triangle_area**, which computes the area of a triangle.

Alternative A:

```
def triangle_area(vertices):
    x, y = vertices
    return 0.5 * abs(x2*y3-x3*y2-x1*y3+x3*y1+x1*y2-x2*y1)
```

Alternative B:

```
def triangle_area(coordinates):
    (x1,y1),(x2,y2),(x3,y3) = coordinates
    return 0.5 * abs(x2*y3-x3*y2-x1*y3+x3*y1+x1*y2-x2*y1)
```

Alternative C:

```
def triangle_area(corners):
    x1, y1 = v1; x2, y2 = v2; x3, y3 = v3
    return 0.5 * abs(x2*y3-x3*y2-x1*y3+x3*y1+x1*y2-x2*y1)
```

Which of the alternatives will make the following test run without error messages?

```
def test_triangle_area():
    v1 = [0,0]; v2 = [1,0]; v3 = [0,2]
    vertices = [v1, v2, v3]
    expected = 1
    computed = triangle_area(vertices)
    tol = 1E-14
    success = abs(expected - computed) < tol
    msg = f"computed area={computed} != {expected}(expected)"
    assert success, msg

test_triangle_area()
```

Select one alternative:

○ Alternative A is correct

○ Alternative B is correct

○ All three alternatives are wrong

○ Alternative C is correct

Maximum marks: 1

12 While-løkke

What is the value of the variable **n** after this code has been run?

```
n = 0
k = 4
while n < k:
    n = n + 2
    k = k + 1
```

Select one alternative:

○ 10

○ 12

○ 8

○ 6

Maximum marks: 1

13 Hva gjør funksjonen

What does this function do? You can assume that the argument **arg** is a list of numbers.

```
def f(arg):
   m = arg[0]
   for e in arg[1:]:
      if e > m:
         m = e
   return m
```

Select one alternative:

○ The function sorts the list arg from the smallest to the largest value

○ The function returns the smallest value in the list arg

○ The function sorts the list arg from the largest to the smallest value

○ The function returns the largest value in the list arg

Maximum marks: 1

14 Finn feilen i koden

In which line will this code stop and print an error message?

```
import numpy as np
from math import sin, pi, exp
t = np.linspace(0,20,101)
x = np.zeros(100)

for i in range(len(t)):
   x[i] = (sin(pi * t[i]) * exp(-t[i]))
```

Select one alternative:

○ x[i] = (sin(pi * t[i]) * exp(-t[i]))

○ x = np.zeros(100)

○ for i in range(len(t)):

○ t = np.linspace(0,20,101)

Maximum marks: 1

15 Lesing fra fil

We have a list of currency exchange rates stored in a file exchange_rates.txt, on the format

USA	1	USD	10.7433
EU	1	EUR	11.4773
Sverige	100	SEK	95.86
Danmark	100	DKK	153.76
Storbrit.	1	GBP	13.2723
Sveits	100	CHF	1197.19
Japan	100	JPY	7.2528

The first column is the name of the country, the second column indicates if the exchange rate is given for 1 or 100 units of the given currency, the third column is the code of the currency, and the fourth column is how many NOK (Norwegian kroner) you need to pay for the given amount of the currency (1 or 100).

We want to write a program that reads such a file and creates two lists; one containing the codes and one containing how many NOK you need to pay for *one* unit of the given currency.
The program below is supposed to conduct this task, but one line is missing.

```
codes = []
rates = []
with open('exchange_rates.txt') as infile:
    for line in infile:
        words = line.split()
        codes.append(words[2])
        # missing line goes here
```

Which of the following lines must be added to make the program work as intended?

Select one alternative:

○ rates.append(float(words[3]))

○ rates.append(float(words[3])/int(words[1]))

○ rates.append(words[3])/words[1])

○ rates.append(float(words[4])/int(words[2]))

Maximum marks: 1

16 Hvilket funksjonskall

The function **bisection(f,a,b,tol)** uses the bisection method to solve a non-linear equation on the form f(x) = 0, on the interval from x=a to x=b.

```
def bisection(f,a,b,tol= 1e-3):
    if f(a)*f(b) > 0:
        print(f'No roots or more than one root in [{a},{b}]')
        return
    m = (a+b)/2
    while abs(f(m)) > tol:
        if f(a)*f(m) < 0:
            b = m
        else:
            a = m
        m = (a+b)/2
    return m
```

We want to use the function to solve the equation
$x^3 + 8x - 10 = 0$
in the interval $-5 < x < 5.$

Which function call is correct?
Select one alternative:

○ sol = bisection(f(x): x**3+8*x-10, -5, 5)

○ sol = bisection(f=x**3+8*x-10, -5, 5, 1e-6)

○ sol = bisection(lambda x: x**3+8*x-10, -5, 5)

○ sol = bisection(x**3+8*x-10, -5, 5, 1e-6)

Maximum marks: 1

17 Hva skrives ut?

What is printed when this code is run?

```
for i in range(3,6,2):
    print(i, end=' ')
    for k in range(1, i):
        print(k, end=' ')
```

The argument **end = ' '** to the print function makes each print end with space (' ') instead of a new line.

Select one alternative:

○ 3 1 2 5 1 2 3 4 6 1 2 3 4 5

○ 3 1 2 5 1 2 3 4

○ 3 0 1 2 5 0 1 2 3 4

○ 3 1 2 3 5 1 2 3 4 5

Maximum marks: 1

18 Plotting

The Python function **pi_approx** in the code below implements the sum

$$f = 4\sum_{k=1}^{n} \frac{(-1)^{k+1}}{2k-1}$$

What does the rest of the code do?

```
import matplotlib.pyplot as plt
from math import pi

def pi_approx(n):
    a = 0
    for k in range(1,n+1):
        a += (-1)**(k+1)/(2*k-1)
    return 4*a

index_list = range(1,51)
a_list = []
for i in index_list:
    a_list.append(pi-pi_approx(i))
plt.plot(index_list,a_list)
plt.show()
```

Select one alternative:

○ The code does not work because we can only plot arrays, not lists.

○ Plot the sum as a function of n, for n=1 to n=50.

○ Plot the terms in the sum, for k=1 to k=50.

○ Plot the terms in the sum, for k=1 to k=51.

○ Plot the difference between the sum and pi, for n=1 to 50.

Maximum marks: 1

19 Hva er riktig?

One of the following statements is correct. Which one?

Select an alternative:

○ Numpy arrays can only be used for storing numbers.

○ Adding two Numpy arrays of length n will result in an array of length 2n.

○ Vectorization means to avoid explicit for-loops in the code.

○ Multiplying two Numpy arrays of length n will result in an array of length n*n.

Maximum marks: 1

20 Hva skrives ut?

The file 'temperature.dat' looks as follows:

```
yyyy  mm   tmax    tmin
            degC    degC
1853   1    6.4     2.7
1854   1    3.2    -1.8
1855   1    7.7    -0.6
1856   1    2.6    -1.5
1857   1    6.8     2.1
```

What is printed by the following code?

```
infile = open('temperature.dat')
for i in range(2):
   infile.readline()
year = 0
tmax = -100
for line in infile:
   words = line.split()
   temp = float(words[2])
   if temp > tmax:
      tmax = temp
      year = int(words[0])

print(year, tmax)
```

Select one alternative:

○ 1855 1 7.7 -0.6

○ 1857 6.8

○ 1855 7.7

○ 1853 6.4

○ 1857 1 6.8 2.1

Maximum marks: 1

21 Hva skrives ut?

What is printed in the terminal window when the following code is run?

```
import sys
A = [['-1','0','1'],['0','0','0'],['10','9','8']]
try:
   b = int(A[2])
except IndexError:
   print(f'A has length {len(A):d}')
   sys.exit(1)
except TypeError:
   print(f'Cannot convert {A[2]} to int')
   sys.exit(1)

print(b)
```

Select one alternative:

○ Cannot convert ['10','9','8'] to int

○ A has length 9

○ Cannot convert ['0','0','0'] to int

○ A has length 3

○ 0

Maximum marks: 1

22 Hvilken funksjon?

We want to implement a function that takes a list as input and returns the largest element. The elements in the list are numbers, which may be positive or negative, and we want the function to support the following use:

```
>>> print(my_max([1,4,5,-1]))
5
>>> print(my_max([-1,-2,-5]))
-1
```

Which of the following implementations of the function **my_max(a)** are correct?

Alternative A:

```
def my_max(a):
    max_found = a[0]
    for i in range(1,len(a)):
        if a[i] > max_found:
            max_found = a[i]
    return max_found
```

Alternative B:

```
def my_max(a):
    max_found = 0
    for e in a:
        if e > max_found:
            max_found = e
    return max_found
```

Alternative C:

```
def my_max(a):
    for i in range(1,len(a)):
        if a[i] > a[i-1]:
            max_found = a[i]
        else:
            max_found = a[i-1]
    return max_found
```

Alternative D:

```
def my_max(a):
    max_found = a[0]
    i = 0
    while i < len(a):
        if a[i] >= max_found:
            max_found = a[i]
        i += 1
    return max_found
```

Please match the values:

	Wrong	Correct
Alternative A	○	○
Alternative D	○	○
Alternative C	○	○
Alternative B	○	○

Maximum marks: 2