ⁱ Forside

UNIVERSITY OF OSLO
Faculty of mathematics and natural sciences

Mid-term exam IN1900, MAT-IN1105, IN-KJM1900, IN-GEO1900
Date: October 12, 2022
Attachments: None
Permitted aids: None

- The exam set has 21 multiple choice questions. The maximum score is 25.
- All code in the question text 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 16 and 17 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 the two questions.
- Questions 18 and 21 have two correct answers. It is given 1 point for each correctly marked answer, -1 for incorrectly marked answers, and 0 for missing answers. The maximum score on the question is 2 and the minimum score is 0. One should always mark at least one answer.

¹ Hva skrives ut?

What is printed in the terminal when the following code is run?
a = 3
b = a + 2
a = 100
print(a,b)

Select one alternative:

○ 100 102

○ An error message

○ 3 5

○ 100 5 ✔

Maximum marks: 1

2 Hva skrives ut?

What is printed when this code is run?
x = [4,5,6]
y = [1,2,3]
print(x + y)

Select one alternative:

○ [5,7,9]

○ [4,5,6,1,2,3]   ✔

○ An error message

○ [1,2,3,4,5,6]

Maximum marks: 1

3 Hva skrives ut?

What is printed when this code is run?

```
x = 0
a = 1
b = 4
if a - b < 0:
    if b != 0:
        x = x + 1
    if b > 0:
        x = x + 2
    elif a > 5:
        x = x + 3
    else:
        x = x + 4
else:
    x = x + 5

print(x)
```

Select one alternative:

- ○ 5

- ○ 2

- ○ 1

- ○ 10

- ○ 3   ✔

Maximum marks: 1

4   Hva skrives ut?

What is printed when this code is run?
import math
r = 1
A = math.pi*r2**
print(type(r),type(A))

Select one alternative:

○ <class 'float'> <class 'float'>

○ <class 'int'> <class 'float'>   ✔

○ 1.0 3.14

○ An error message

Maximum marks: 1

What is printed when this code is run?
import math
r = 1
A = math.pi*r2**
print(type(r),type(A))

5 Hva skrives ut?

What is printed when this code is run?

```
a = []
x = []
for n in range(100):
    x.append(n)
    a.append(x)
print(len(a), len(a[-1]))
```

Select one alternative:

- ○ 100 100 ✔
- ○ 101 101
- ○ An error message
- ○ 101 1
- ○ 100 1

Maximum marks: 1

6 Hva skrives ut?

What is printed when this code is run?

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

Select one alternative:

- ○ xam! ✔
- ○ Welcome to the midterm exam!
- ○ An error message
- ○ exam!

Maximum marks: 1

7 Hva skrives ut?

What is printed when this code is run?
```
s = 0
for i in range(2,8,2):
    s += i
print(s)
```

Select one alternative:

○ 22

○ 0

○ 20

○ 12 ✔

Maximum marks: 1

8 Hva skrives ut?

What is printed when this code is run?
```
def f(x,y):
    return 3*x**2 - 4*y

y = 3
x = 2
print(f(1,2))
```

Select one alternative:

○ -5 ✔

○ An error message

○ 0

○ 10

Maximum marks: 1

9 Hva skrives ut?

What is printed when this code is run?
from math import sqrt

def norm(v):
  v_squared = [e2 for e in v]**
  return sqrt(sum(v_squared))

def test_norm():
  v = (4,3)
  expected = 5
  computed = norm(v)
  tol = 1e-6
  msg = f'expected {expected}, got {computed}'
  assert abs(expected -computed) < tol, msg

test_norm()

The function **sum**, which is used inside **norm**, is a built-in function in Python. It takes a list, array, or similar object as argument, and returns the sum of its elements.
Select one alternative:

○ False

○ AssertionError: expected 5, got 0

○ Nothing is printed ✔

○ True

Maximum marks: 1

10 Hva skrives ut?

What is printed when this code is run?

```
F = 0
C1 = []; C2 = []

while F <= 100:
    C1.append(5/9 * (F-32))
    F += 10

for F in range(0,100,10):
    C2.append(5/9 * (F-32))

print(F, len(C1) == len(C2))
```

Select one alternative:

○ 90 True

○ 100 True

○ 100 False

○ 90 False ✔

Maximum marks: 1

11 I hvilken linje feiler koden?

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

n = 100
x = [0]*(n+1)
dx = 1.0/n
for i in range(n+1):
    x[i+1] = i * dx

Select one alternative:

○ for i in range(n+1):

○ x = [0]*(n+1)

○ dx = 1.0/n

○ x[i+1] = i * dx   ✔

Maximum marks: 1

12 I hvilken linje feiler koden?

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

```
def g(x):
    return x**3 + x**2

def f(x,y):
    return x + 2*y

x = 1
print(f(x,g(x)))
print(g(y))
```

Select one alternative:

○ return x**3 + x**2

○ print(f(x,g(x)))

○ print(g(y))   ✔

○ return x + 2*y

Maximum marks: 1

13 Hvilken linje mangler?

Which line must be added to this code for the test to pass?

```
def triangle_area(corners):
    #insert missing line here
    return 0.5*abs(x2*y3-x3*y2-x1*y3+x3*y1+x1*y2-x2*y1)


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:

○ (x1,y1),(x2,y2),(x3,y3) = corners   ✔

○ x1, y1, x2, y2, x3, y3 = zip(corners)

○ x1,y1 = v1; x2,y2=v2; x3,y3=v3

○ x, y = corners

Maximum marks: 1

14 Hva er riktig?

Which of the following statements is/are correct?
1. A function must always include a return statement.
2. A for loop can always be written as a while loop.
3. A while loop will usually give shorter and simpler code than a for loop.

Select one alternative:

○ Statement 2 is correct. The rest are wrong. ✔

○ Statements 1 and 3 are correct. Statement 2 is wrong.

○ Statements 1 and 2 are correct. Statement 3 is wrong.

○ Statement 1 is correct. The rest are wrong.

Maximum marks: 1

15 Hvilken linje mangler?

A file "oxygen.txt" contains information about isotopes of oxygen. The file has the following contents:

Isotope	Weight [g/mol]	Natural abundance
(16)O	15.99491	0.99759
(17)O	16.99913	0.00037
(18)O	17.99916	0.00204

We want the code below to read the file, calculate the molar mass of oxygen, and print the result to the screen. The molar mass is the sum $\sum_i m_i w_i$, where m_i is the mass (Weight) of an isotope and w_i is the natural abundance. Which line must be added for the code to work?

```
with open('oxygen.txt') as infile:
    s = 0
    infile.readline()
    for line in infile:
        #insert missing line here
        s += m*w

print(s)
```

Select one alternative:

○ m, w = float(line.split()[1:])

○ m, w = [float(w) for w in line.split()[1:]] ✔

○ m, w = float(line)

○ m, w = float(line[1]), float(line[2])

Maximum marks: 1

16 Lister og boolske uttrykk

What are the values of the boolean expressions after the following code has been run?

```
def add(a,b):
    return [a_i+b_i for a_i,b_i in zip(a,b)]


x = list(range(5))
y = [2,3,4]
u = add(x,y)
v = x+y
```

Please match the values:

	True	Undefined/gives error	False
u[0] == x[0]+y[0]	○ ✔	○	○
u == v	○	○	○ ✔
len(u) > len(v) and u[-1] > v[-1]	○	○	○ ✔
len(u) == len(v)	○	○	○ ✔

Maximum marks: 2

17 Boolske uttrykk

What are the values of the boolean expressions?
Please match the values:

	True	Udefinert/gir feil	False
True and (False and True)	○	○	○ ✔
True and (False or True)	○ ✔	○	○
False or 4 > 3	○ ✔	○	○
not True or False	○	○	○ ✔

Maximum marks: 2

18 Hvilke koder er riktige?

The standard deviation of a sequence of numbers x_i, $i = 1, ...,N$ is defined by

$$s_N = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})^2}$$

where $\bar{x}$ is the mean value of the numbers.

Assume that we have already implemented a function **mean(x_list)** which computes the mean value of a list of numbers. We now want to implement a function **std** which computes and returns the standard deviation of the numbers in the list. Two of the following alternatives are correct. Which ones?

Alternative A:
```
def std(x_list):
    s = 0
    n = len(x_list)
    for i in range(n+1):
        s += (x_list[i]-mean(x_list[i]))**2
    return(sqrt(s/n))
```

Alternative B:
```
def std(x_list):
    s = 0
    m = mean(x_list)
    for x in x_list:
        s += (x-m)**2
    return(sqrt(s/len(x_list)))
```

Alternative C:
```
def std(x):
    s = 0
    m = mean(x)
    n = len(x)
    for i in range(n):
        s += (x[i]-m)**2
    return(sqrt(s/n))
```

Alternative D:
```
def std(x_list):
    s = 0
    m = mean(x_list)
    for i in range(len(x_list)-1):
        s += (x[i]-m)**2
    return(sqrt(s/len(x_list)))
```

One (1) point for each correctly marked answer, minus one (-1) point for each wrongly marked answer. Maximum two points total. The minimum score on the question is zero, so you should always mark at least one answer.

Select two alternatives:

☐ Alternative C is correct ✔

☐ Alternative B is correct ✔

☐ Alternative D is correct

☐ Alternative A is correct

Maximum marks: 2

19 Hva gjør funksjonen?

What does this function do? Assume that the argument **numbers** is a list of numbers.

```
def my_fun(numbers):
    n = len(numbers)
    for i in range(n):
        for j in range(n-1):
            if numbers[j] < numbers[j+1]:
                tmp = numbers[j]
                numbers[j] = numbers[j+1]
                numbers[j+1] = tmp
    return numbers
```

Select one alternative:

○ Returns the same list of numbers, with the order unchanged.

○ Returns a list with the numbers sorted in decreasing order. ✔

○ Returns a list with the numbers sorted in random order.

○ Returns a list with the numbers sorted in increasing order.

Maximum marks: 1

20   Hva skrives ut?

The file "temp_oct_1945.txt" has the following contents:
Year: 1945. Month: October. Location: Blindern(Oslo).
7.2   8.1 8.9 11.6 7.7 8.7   6.9
5.4   8.8 8.9 3.7   3.3   5.2   9.6
10.8 5.0 5.4   9.5   5.3 5.8   2.3
4.1 6.6 8.2   6.1   8.9 6.6   4.1
2.8   2.1 4.1

What is printed when the following code is run? You can assume that the code file and the data file are located in the same directory.

```python
def extract_data(filename):
    with open(filename) as infile:
        infile.readline()
        temp = []
        for line in infile:
            numbers = [float(t) for t in line.split()]
            temp += numbers
    return temp


def analyze(numbers):
    count = len(numbers)
    sum = 0
    for n in numbers:
        sum += n
    mean = sum/count
    return count, sum, mean


print(analyze(extract_data('temp_oct_1945.txt')))
```

Select one alternative:

○ (31, 201.7, 6.506451612903225)   ✔

○ En feilmelding

○ [7.2, 8.1, 8.9, 11.6, 7.7, 8.7, 6.9, 5.4, ..., 8.2, 6.1, 8.9, 6.6, 4.1, 2.8, 2.1, 4.1]

○ October 1945 = 6.506

Maximum marks: 1

21 Hvilken kode er riktig?

The file "marathons.txt" contains a collection of marathon times recorded in the period 1981-2022:

2:05:42; Khalid Khannouchi; 1999;   Chicago
2:05:38; Khalid Khannouchi; 2002; London
2:04:55; Paul Tergat; 2003; Berlin
2:07:12; Carlos Lopes; 1985; Rotterdam
2:04:26; Haile Gebrselassie; 2007; Berlin
2:03:59; Haile Gebrselassie; 2008; Berlin
2:06:50; Belayneh Dinsamo; 1988; Rotterdam
2:06:05; Ronaldo da Costa; 1998; Berlin
2:03:23; Wilson Kipsang;   2013;   Berlin
2:02:57; Dennis Kimetto; 2014; Berlin
2:08:18; Robert De Castella; 1981; Fukuoka
2:01:39; Eliud Kipchoge; 2018; Berlin
2:01:09; Eliud Kipchoge; 2022; Berlin
2:08:05; Steve Jones; 1984;   Chicago
2:03:38; Patrick Makau; 2011; Berlin

We want to write a code which reads such a file and prints the fastest time as well as the name of the person who got it. The output from the code shall look as follows:

2:01:09 Eliud Kipchoge

Two of the alternatives below are correct and will give this output. Which ones?

Alternative A:
```
fastest = (14400,'4:00:00')
name = 'No record found'
with open('marathons.txt') as infile:
    for line in infile:
        w = line.split(';')
        h,m,s = [float(ti) for ti in w[0].split(':')]
        seconds = h*3600 + m*60 + s
        if seconds < fastest[0]:
            fastest = (seconds,w[0])
            name = w[1]
print(fastest[1], name)
```

Alternative B:
```
fastest = 100000
name = 'No record found'
with open('marathons.txt') as infile:
    for line in infile:
        w = line.split(';')
        h,m,s = [float(ti) for ti in w[0].split(':')]
        seconds = h*3600 + m*60 + s
        if seconds < fastest:
            fastest = seconds
            name = w[1]
print(fastest, name)
```

Alternative C:

```
fastest = (14400, '4:00:00')
name = 'No record found'
with open('marathons.txt') as infile:
    for line in infile:
        w = line.split()
        h,m,s = [float(ti) for ti in w[0].split(':')]
        seconds = h*3600 + m*60 + s
        if seconds < fastest[0]:
            fastest = (seconds,w[0])
            name = w[1]
print(fastest[1], name)
```

Alternative D:

```
fastest = 14400
name = 'No record found'
with open('marathons.txt') as infile:
    for line in infile:
        w = line.split(';')
        h,m,s = [float(ti) for ti in w[0].split(':')]
        seconds = h*3600 + m*60 + s
        if seconds < fastest:
            fastest = seconds
            record = w
print(record[0],record[1])
```

One (1) point for each correctly marked answer, minus one (-1) point for each wrongly marked answer. Maximum two points total. The minimum score on the question is zero, so you should always mark at least one answer.

Select one or more alternatives:

- ☐ Alternative C is correct

- ☐ Alternative A is correct ✔

- ☐ Alternative D is correct ✔

- ☐ Alternative B is correct

Maximum marks: 2