UNIVERSITETET I OSLO

KANDIDAT

125007

PRØVE

JUS5080 1 Programming for lawyers

Emnekode	JUS5080
Vurderingsform	Individuell skriftlig prøve
Starttid	20.12.2024 08:00
Sluttid	20.12.2024 11:00
Sensurfrist	09.01.2025 22:59
PDF opprettet	24.01.2025 08:23

Seksjon 1

Oppgave	Tittel	Oppgavetype
ℹ	Information	Informasjon eller ressurser
1	if 1 H24	Fyll inn tekst
2	If 2 H24	Fyll inn tekst
3	if 3 H24	Fyll inn tekst
4	H24 and/or	Fyll inn tekst

Seksjon 2

Oppgave	Tittel	Oppgavetype
5	H24 lists 1	Fyll inn tekst
6	H24 List 2	Fyll inn tekst
7	H24 Nested Lists	Fyll inn tekst

Seksjon 3

Oppgave	Tittel	Oppgavetype
8	H24 dictionaries 1	Fyll inn tekst
9	H24 Dictionaries 2	Programmering
10	H24 dictionaries 3	Fyll inn tekst

Seksjon 4

Oppgave	Tittel	Oppgavetype
11	H24 Loop 1	Fyll inn tekst
12	H24 Read file	Programmering

Seksjon 5

Oppgave	Tittel	Oppgavetype
13	H24 Functions 1	Programmering
14	H24 Find sum	Programmering
15	H24 Find word	Programmering

1 if 1 H24

Enter a number in the box below, so that "Okay" will be printed:

```
number = 3
number = number + 4
if number == 7:
    print('Okay')
```

2 If 2 H24

Complete the code below, so that s*maller* is printed.

```
number = 3
if number < 17:
    print('smaller')
else:
    print('greater')
```

3 if 3 H24

Complete the code below, so that the text *JUS 5080* is printed:

```
a = 5080

if False:

    a = a + 2

print('JUS', a)
```

4 H24 and/or

Complete the code below with boolean operators (and, or, not) so that *okay* is printed in each case.

```
a = 1
b = 4


if a == 5 or b > 0:

   print('okay')


if b > 10 or a == 2:

   print('not this')
else:
   print('okay')


if b >= 2 and a < 0:

   print('not this')
else:
   print('okay')


if False or a >= 0:

   print('okay')
```

5 H24 lists 1

Complete the code below so that 'fun' is printed.

```
words = ['law', ' is', ' fun']
print(words[ 2 ])
```

6 H24 List 2

Complete the code below so that 5 is printed.

```
numbers = [5, 2, 4, 3]
print(numbers[3] + numbers[ 1 ])
```

Complete the code below so that 7 is printed.

```
numbers = [5, 2, 6, -2, 1, 3]
print(numbers[-2] + numbers[ 2 ])
```

7 H24 Nested Lists

```
numbers = [[0, 1, 2, 3, 5, 6, 7],
          [-5, -3, 0, 1, 2, 3, 5, 6],
          [1, 2, 3, 5, 6, 7],
          [10, 11, 2, 3, 5, 6, 7],
          [0, 8, 9, 10, 11, 12, 13, 14, 15]]
```

What is the value of the expression **numbers[0][2]** ?

2

What is the value of the expression **numbers[1][-2]** ?

5

What is the value of the expression **numbers[4][1] + numbers[1][3]** ?

9

What is the value of the expression **numbers[2][1] - numbers[0][1]** ?

3

What is the value of the expression **numbers[3][-2] - numbers[0][-3]** ?

1

8 H24 dictionaries 1

Create a dictionary containing the client list below. The keys should be the clients' names, and the values should be the client numbers.

Client Name	Client Number
Sonic	1
Zelda	4

client_list = { "Sonic": 1, "Zelda": 4 }

9 H24 Dictionaries 2

We have a dictionary that contains some membership fees that are due. Write code to calculate the total amount due by getting the values from the dictionary.You should not simply type the actual amounts (numbers) in your code.

membership_fees = {
'Jane': 10,
'Sonic': 75,
'Zelda': 25,
'Mario': 15
}

Fill in your answer here

```
total_amount_due = 0
for key in membership_fees:
    total_amount_due += membership_fees[key]

# There is syntax for doing something like "for key, value in dict", as well as doing
    I'm not 110% what the syntax is, and it also makes the code less readable, so her
```

10 H24 dictionaries 3

We first execute this code:

```
dictionary = {'a': [0, 1, 2, 3, 4, 5],
          'bb': [8, 9, 12, 13, 14],
          'g': [1, 2, 3, 5, 6, 7],
          'hi': [0, 2, 3, 5, 6],
          }
dictionary['g'].append(7)
dictionary['e'] = [2, 3, 5, 6]
```

Now, what are the outputs of the following statements?

```
print(dictionary['hi'][3])
```

5

```
print(len(dictionary['hi']))
```

5

```
print(len(dictionary['g']))
```

7

```
print(len(dictionary))
```

5

```
dictionary['g'] = ['hi']
dictionary['b'] = 15
print(len(dictionary['g']))
```

1

```
dictionary['hi'].pop(0)
print(len(dictionary['hi']))
```

4

11 H24 Loop 1

Complete the code below so that Programming for lawyers is printed.

for token in ['Programming', 'for', 'lawyers']:

print (token)

12 H24 Read file

You have a text file called data.txt that you need to open, read and print line by line. Write some code to do this, without using pandas. You don't need to handle any exceptions, but remember to close the file.

Fill in your answer here

```
with open('data.txt') as file:
    for line in file.read():
        print(line)
```

13 H24 Functions 1

Write a function called message that always prints "Law is fun"

Fill in your answer here

```
def message():
    print("Law is fun")
```

14 H24 Find sum

Python has a built-in function sum(), which calculates the total sum of a list of numbers. Your task is to write your own version of this function, without using Python’s sum() function.

Make a function called my_sum, which takes a list of numbers as its only argument. The function should return the sum you get when adding all the numbers.

When you have created the function, call it with this list as an argument, and print the result: [5, 4, 3]

(The function should work for any list of numbers, not just this example.)

Fill in your answer here

```
# Takes a list of numbers an an argument
# Returns the sum of the numbers
def my_sum(list_of_numbers):
    total = 0
    for number in list_of_numbers:
        total += number
    return total

print(my_sum([5,4,3]))

# Again, there is some syntax for doing the operation within this function all on o
    to read, and I'm not 110% sure how to do it.
```

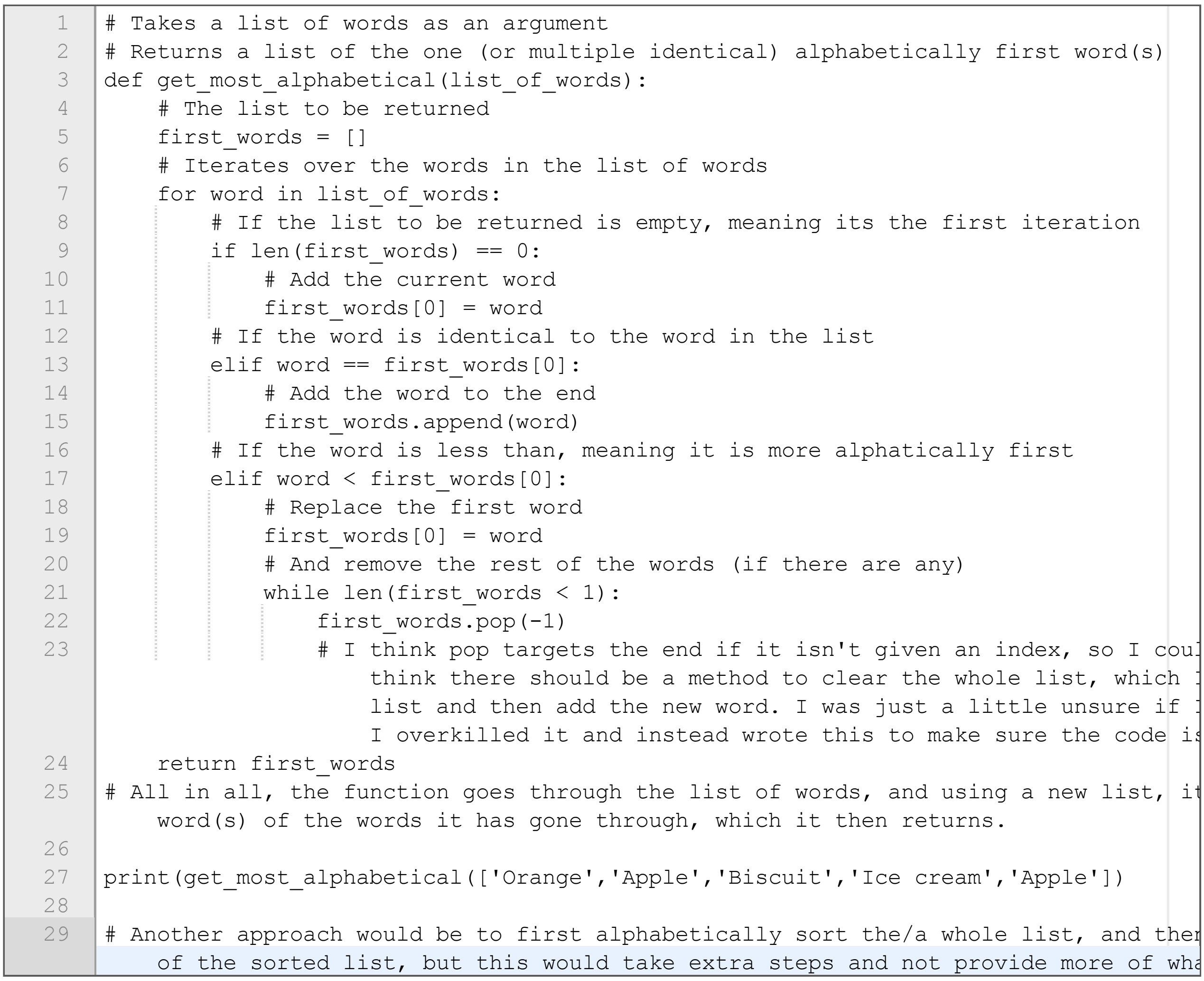
15 H24 Find word

Write a function that takes a list of words as its argument. The function should find the word(s) that comes alphabetically first and return it/them in a list. If multiple words are tied for the position as alphabetically first, i.e. the same word occurs more than once, they should all be in the list that is returned.

When you have created the function, call it with this list as an argument and print the result: ['Orange', 'Apple', 'Biscuit', 'Ice cream', 'Apple']. The correct result for this list is ['Apple', 'Apple'], since 'Apple' occurs twice.

(The function should work for any list of words, not just this example.)

Fill in your answer here



```
# Takes a list of words as an argument
# Returns a list of the one (or multiple identical) alphabetically first word(s)
def get_most_alphabetical(list_of_words):
    # The list to be returned
    first_words = []
    # Iterates over the words in the list of words
    for word in list_of_words:
        # If the list to be returned is empty, meaning its the first iteration
        if len(first_words) == 0:
            # Add the current word
            first_words[0] = word
        # If the word is identical to the word in the list
        elif word == first_words[0]:
            # Add the word to the end
            first_words.append(word)
        # If the word is less than, meaning it is more alphatically first
        elif word < first_words[0]:
            # Replace the first word
            first_words[0] = word
            # And remove the rest of the words (if there are any)
            while len(first_words < 1):
                first_words.pop(-1)
                # I think pop targets the end if it isn't given an index, so I coul
                    think there should be a method to clear the whole list, which I
                    list and then add the new word. I was just a little unsure if I
                    I overkilled it and instead wrote this to make sure the code is
    return first_words
# All in all, the function goes through the list of words, and using a new list, it
    word(s) of the words it has gone through, which it then returns.

print(get_most_alphabetical(['Orange','Apple','Biscuit','Ice cream','Apple'])

# Another approach would be to first alphabetically sort the/a whole list, and then
    of the sorted list, but this would take extra steps and not provide more of wha
```