MATEMATISKA INSTITUTIONEN
STOCKHOLMS UNIVERSITET
Avd. Beräkningsmatematik
Kursledare: Lars Arvestad
Examinator: Lars Arvestad

Tentamensskrivning i
DA2005 Programming Techniques
7.5 hp
2024-04-11

- The exam includes multiple-choice questions where at least one option is correct. If you answer incorrectly or do not provide the exact number of correct alternatives, you will receive zero points for the question.
- You must pass Part A (4 correct out of 8 questions) for Part B to be graded.
- Part B consists of questions with varying points (total of 12p).
- No `import` statements (Python's standard library or external libraries) are allowed unless mentioned or included in the task, but you may use built-in functions such as `len`, `range`, and `map`.
- All code refers to **Python 3**, not e.g., Python 2.7.
- **Tools:** An A4 with as much information as you wish. You may write on both sides.
- **Grade boundaries:** E: 10, D: 12, C: 14, B: 16, A: 18, out of a maximum of 20.

Part A: multiple-choice questions

Please collect the answers to Part A on a separate answer sheet.

Obs, tentan finns på svenska på sidan 6 och framåt!

1. Which list is returned by the function `fcn`?

```
def fcn():
    res = []
    for i in range(5):
        res.append(i**2)
    return res
```

A. `[]`
B. `[1, 2, 3, 4, 5]`
C. `[2, 4, 6, 8, 10]`
D. `[1, 4, 9, 16, 25]`
E. None of the above, they are all wrong!

2. What is printed by the code on the right?

```
d = {
    'Amiral': 'insekt',
    'Sorgmantel': 'insekt',
    'Storsjöodjuret': 'folktro'
}
print(d['Amiral'][1])
```

A. `'i'`
B. `'n'`
C. `'insekt'`
D. `'folktro'`
E. An error message

3. What is the result of the following expression?

```
list(map(lambda x: chr(ord(x)+1), 'SU'))
```

The function `ord` returns an integer representing a character. For example, `ord('A')` returns the value 65, `ord('B')` returns the value 66, and so on. The function `chr` returns a character given an integer (according to Unicode). For example, `chr(65)` returns the character A.

A. `'US'`
B. `'VT'`
C. `['S', 'U']`
D. `['T', 'V']`
E. An error message

4. What is printed by `print(f(a))` given the code on the right?

A. 4

B. 5

C. 7

D. 9

E. An error message

```
a = 1

def f(a=2, b=3):
    return g(a, b)

def g(a=4, b=5):
    return a + b
```

5. What is printed by the code on the right?

A. `'Sun'`

B. `['Sun', 'Bathing', 'Sweat', 'Allergies']`

C. One string at a time of 'Sun', 'Bathing', 'Sweat', and 'Allergies'

D. Endlessly repeating `'Sun'`

E. Endlessly repeating `['Sun', 'Bathing', 'Sweat', 'Allergies']`

F. An error message

```
def summer():
    lst = ['Sun', 'Bathing',
           'Sweat', 'Allergies']
    while lst:
        print(lst[0])
        lst = lst[1:]
summer()
```

6. The constant π (i.e., 3.1415...) is defined in the standard module `math`. In what ways can one make the statement `print(pi)` work?

A. `import math`

B. `import math as m`

C. `import math as pi`

D. `from math import *`

E. `from math import pi`

7. Consider the code on the right. What is returned by `computation([1,2,3,4])`?

A. 0

B. 1

C. 4

D. 5

E. 10

```
def computation(lst):
    if len(lst) > 0:
        return lst[0] + computation(lst[1:])
    else:
        return 0
```

8. Consider the code on the right. What is returned by `thefunction('jokkmokk')`?

A. `'' # empty string`

B. `'jokm'`

C. `'jokmok'`

D. `'jkkmkk'`

E. `'jokkmokk'`

```
def thefunction(s):
    d = {}
    for x in s:
        if x in d:
            d[x] += 1
        else:
            d[x] = 1

    r = ''
    for x in s:
        if x in d:
            r += x
            del d[x]
    return r
```

Part B: coding questions

Please use one sheet of paper for each question in Part B.

9. (2p) Write the function `average(lst)` that takes a list of numbers as input and returns the arithmetic mean of the numbers. The arithmetic mean is $\frac{1}{n}\sum_{i=1}^{n} x_i$ if the list contains the numbers $x_1, x_2, \ldots, x_n$. If the list is empty, the function should raise a `ValueError`.

You are not allowed to use the `mean` function from the module "statistics" (or similar if available in other standard modules). **Example usage:**

```
[In: ] average([1])
[Out:] 1.0
[In: ] average([1, 3])
[Out:] 2.0
[In: ] average([])
[Out:]
Traceback (most recent call last):
   [COMMENT: Details removed by examiner]
ValueError
```

10. (2p) Write the function `get_user_int(a, b)` which asks a user for an integer in the interval a to b (i.e., $[a, b]$). The function should continue asking until the user responds with an integer in the interval, and then return that integer. If the user provides an integer outside the interval, the message "That was outside the interval" should be displayed. If the user responds with something that cannot be interpreted as an integer, the message "Please enter an integer" should be displayed.

Example usage:

```
[In: ] get_user_int(0,1)
[Out:]
Which integer? a
Please enter an integer.
Which integer? 2
That was outside the interval.
Which integer? 0
0
[In: ] get_user_int(1,10)
[Out:]
Which integer? 0
That was outside the interval.
Which integer? 11
That was outside the interval.
Which integer? 5
5
```

In the example, the function returns first 0 and then 5.

11. (2p) Write the function `count_ampersands(filename)` that counts and returns the number of lines in the file *filename* containing an ampersand (the & character). If the file does not exist, there should be a `FileNotFoundError`.

Example usage:

Assume that the file "exam.txt" contains the text for this exam, "empty.txt" is an empty file, and the file "three.txt" looks like this:

```
&
Adam & Eve

We are looking for this character: &
```

Note that this exam contains the character four times. The function should work like this:

```
[In: ] count_ampersands('exam.txt')
[Out:] 4
[In: ] count_ampersands('empty.txt')
[Out:] 0
[In: ] count_ampersands('three.txt')
[Out:] 3
```

12. (2p) The function below does not work as the documentation string says it should. Correct the function without introducing a new algorithmic idea.

Note: the code for this problem has been edited since the exam. Originally, the *correct* code was included rather than this buggy code.

```
def increasing(lst):
    '''
    `lst` must be a list of numbers.

    Return True if the list `lst` is monotonically increasing,
    and False otherwise. A list shorter than 2 is
    considered increasing.
    '''
    if len(lst) < 2:
       return True
    for i in range(lst):
        if lst[i] > lst[i+1]:
            return True
    return False
```

13. (1p) Assume that the function `increasing` (from question 12) works correctly and that there is a corresponding correct function `decreasing` that determines whether a list is monotonically decreasing. Use these two functions to write the function `monotonic(lst)` that returns `True` if `lst` is monotonous and otherwise `False`.

Requirements:

- You must call the functions `increasing` and `decreasing` in your solution.
- You should not re-implement the code for `increasing` and `decreasing` in your solution.
- You are not allowed to use list comprehensions or **`sorted`**, as used in two example solutions in the previous exam.

Example usage:

```
[In: ] monotonic([1, 2, 3, 4])
[Out:] True
[In: ] monotonic([4, 3, 2, 1])
[Out:] True
[In: ] monotonic([1, 9, 2, 7])
[Out:] False
[In: ] monotonic([0, 0, 0, 0, 0])
[Out:] True
```

A series is monotonic if it either increases or decreases: given a list $L = (l_0, l_1, \ldots, l_n)$, either $l_i \leq l_{i+1}$ for $0 \leq i < n$ or $l_i \geq l_{i+1}$ for $0 \leq i < n$.

14. (3p) Write a class 'Exercise' that can register and display how much time you spend on various forms of exercise. The class should have two methods:

- With the method `register`, you input exercise data in the form of an activity and the number of minutes you spent on the activity.
- The method `show` prints out a summary of your exercise, with a simple heading and then one row per activity and the total number of minutes recorded for the activity.

Your class should be implemented so that the examples below work.

Example usage:

```
[In: ] workout = Exercise()
[In: ] workout.register('running', 20)
[In: ] workout.register('badminton', 60)
[In: ] workout.register('running', 30)
[In: ] workout.register('swimming', 25)
[In: ] workout.register('running', 15)
[In: ] workout.show()
[Out:]
Activity  Duration (min)
running         65
badminton       60
swimming        25
[In: ] homer = Exercise()
[In: ] homer.register('walked', 1)
[In: ] homer.show()
[Out:]
Activity  Duration (min)
walked           1
```

In the example, the columns are neatly aligned, but you do not need to implement that.

The exam in Swedish

- Tentan har flervalsfrågor där minst ett svarsalternativ är korrekt. Om man svarar fel eller inte har exakt antal rätta alternativ får man noll poäng på frågan.
- Man måste bli godkänd på del A (4 rätt på 8 frågor) för att del B ska rättas.
- Del B består av frågor med varierande poäng (totalt 12p).
- Inga `import` (Pythons standardbibliotek eller externa bibliotek) får användas om de inte nämns eller finns med i uppgiften. Man får dock använda inbyggda funktioner som `len`, `range` och `map`.
- All kod avser **Python 3**, dvs *inte* t.ex. Python 2.7
- **Hjälpmedel:** Ett A4 med så mycket information du vill. Du får skriva på båda sidorna.
- **Betygsgränser:** E: 10, D: 12, C: 14, B: 16, A: 18, av maximala 20.

Del A: flervalsfrågor

Var snäll samla svaren på del A på ett svarspapper.

1. Vilken lista returneras av funktionen `fcn`?

```
def fcn():
    res = []
    for i in range(5):
        res.append(i**2)
    return res
```

A. `[]`
B. `[1, 2, 3, 4, 5]`
C. `[2, 4, 6, 8, 10]`
D. `[1, 4, 9, 16, 25]`
E. Ingen av de ovanstående, de är alla fel!

2. Vad skrivs ut av koden till höger?

```
d = {
    'Amiral': 'insekt',
    'Sorgmantel': 'insekt',
    'Storsjöodjuret': 'folktro'
}
print(d['Amiral'][1])
```

A. `'i'`
B. `'n'`
C. `'insekt'`
D. `'folktro'`
E. Ett felmeddelande

3. Vad är resultatet av följande uttryck?

```
list(map(lambda x: chr(ord(x)+1), 'SU'))
```

Funktionen `ord` returnerar ett heltal som representerar en bokstav. Till exempel returnerar `ord('A')` värdet 65, `ord('B')` värdet 66, o.s.v. Funktionen `chr` returnerar en bokstav givet ett heltal (enligt Unicode). Till exempel ger `chr(65)` bokstaven A.

A. `'US'`
B. `'VT'`
C. `['S', 'U']`
D. `['T', 'V']`
E. Ett felmeddelande

4. Vad skrivs ut av `print(f(a))` givet koden till höger?

```
a = 1

def f(a=2, b=3):
    return g(a, b)

def g(a=4, b=5):
    return a + b
```

A. 4
B. 5
C. 7
D. 9
E. Ett felmeddelande

5. Vad skrivs ut av koden till höger?

A. `'Sun'`

B. `['Sun', 'Bathing', 'Sweat', 'Allergies']`

C. En sträng i taget av 'Sun', 'Bathing', 'Sweat', och 'Allergies'

D. Oändligt upprepande av `'Sun'`

E. Oändligt upprepande av `['Sun', 'Bathing', 'Sweat', 'Allergies']`

F. Ett felmeddelande

```
def summer():
    lst = ['Sun', 'Bathing',
           'Sweat', 'Allergies']
    while lst:
        print(lst[0])
        lst = lst[1:]
summer()
```

6. Konstanten π (d.v.s. 3.1415...) finns definierad i standardmodulen `math`. På vilka sätt kan man få satsen `print(pi)` att fungera?

A. `import math`

B. `import math as m`

C. `import math as pi`

D. `from math import *`

E. `from math import pi`

7. Betrakta koden till höger. Vad returneras av `computation([1,2,3,4])`?

A. 0

B. 1

C. 4

D. 5

E. 10

```
def computation(lst):
    if len(lst) > 0:
        return lst[0] + computation(lst[1:])
    else:
        return 0
```

8. Betrakta koden till höger. Vad returneras av `thefunction('jokkmokk')`?

A. `'' # tomma strängen`

B. `'jokm'`

C. `'jokmok'`

D. `'jkkmkk'`

E. `'jokkmokk'`

```
def thefunction(s):
    d = {}
    for x in s:
        if x in d:
            d[x] += 1
        else:
            d[x] = 1

    r = ''
    for x in s:
        if x in d:
            r += x
            del d[x]
    return r
```

Del B: kodfrågor

Var snäll använd ett papper till varje fråga i del B.

9. (2p) Skriv funktionen `average(lst)` som tar en lista med tal som argument och returnerar talens aritmetiska medelvärde. Det aritmetiska medelvärdet är $\frac{1}{n}\sum_{i=1}^{n} x_i$ om listan innehåller talen $x_1, x_2, \ldots, x_n$.

Om listan är tom ska funktionen ge särfallet `ValueError`.

Du får inte använda funktionen `mean` från modulen "statistics" (eller liknande om det finns i andra standardmoduler). **Exempelanvändning:**

```
[In: ] average([1])
[Out:] 1.0
[In: ] average([1, 3])
[Out:] 2.0
[In: ] average([])
[Out:]
Traceback (most recent call last):
   [COMMENT: Details removed by examiner]
ValueError
```

10. (2p) Skriv funktionen `get_user_int(a, b)` som frågar en användare efter ett heltal i intervallet a till b (d.v.s. $[a, b]$). Funktionen ska fortsätta fråga tills användaren svarar med ett heltal i intervallet och då returneras heltalet. Om användaren ger ett heltal utanför intervallet så ska meddelandet "That was outside the interval" skrivas ut. Om användaren svarar med något som inte kan tolkas om till ett heltal ska meddelandet "Please enter an integer" skrivas ut.

Exempelanvändning:

```
[In: ] get_user_int(0,1)
[Out:]
Which integer? a
Please enter an integer.
Which integer? 2
That was outside the interval.
Which integer? 0
0
[In: ] get_user_int(1,10)
[Out:]
Which integer? 0
That was outside the interval.
Which integer? 11
That was outside the interval.
Which integer? 5
5
```

I exemplet returneras först 0 och sedan 5 av funktionen.

11. (2p) Skriv funktionen `count_ampersands(filename)` som räknar och returnerar antalet rader i filen *filename* som innehåller ampersand (tecknet &). Om filen inte finns ska det bli ett särfall `FileNotFoundError`.

Exempelanvändning:

Antag att filen "exam.txt" innehåller texten för denna tenta, "empty.txt" är en tom fil, och filen "three.txt" ser ut så här:

```
&
Adam & Eve

We are looking for this character: &
```

Observera att denna tenta innehåller tecknet fyra gånger.

Funktionen ska fungera så här:

```
[In: ] count_ampersands('exam.txt')
[Out:] 4
[In: ] count_ampersands('empty.txt')
[Out:] 0
[In: ] count_ampersands('three.txt')
[Out:] 3
```

12. (2p) Funktionen nedan fungerar inte som dokumentationssträngen säger att den ska göra. Korrigera funktionen utan att införa en ny algoritmisk idé.

```
def increasing(lst):
    '''
    `lst` must be a list of numbers.

    Return True if the list `lst` is monotonically increasing,
    and False otherwise. A list shorter than 2 is
    considered increasing.
    '''
    if len(lst) < 2:
       return True
    for i in range(lst):
        if lst[i] > lst[i+1]:
            return True
    return False
```

13. (1p) Antag att funktionen `increasing` (från fråga 12) fungerar korrekt och att det finns en motsvarande korrekt funktion `decreasing` som avgör om en lista är monotont avtagande. Använd dessa två funktioner till att skriva funktionen `monotonic(lst)` som returnerar `True` om `lst` är monoton och annars `False`.

Krav:

- Du måste anropa funktionerna `increasing` och `decreasing` i din lösning.
- Du ska inte åter-implementera koden för `increasing` och `decreasing` i din lösning.
- Du får inte använda listomfattning eller **`sorted`**, som användes i två exempellösningar i förra tentan.

Exempelanvändning:

```
[In: ] monotonic([1, 2, 3, 4])
[Out:] True
[In: ] monotonic([4, 3, 2, 1])
[Out:] True
[In: ] monotonic([1, 9, 2, 7])
[Out:] False
[In: ] monotonic([0, 0, 0, 0, 0])
[Out:] True
```

En serie är monoton (eng: *monotonic*) om den antingen växer eller avtar: givet en lista $L = (l_0, l_1, \ldots, l_n)$ är antingen $l_i \leq l_{i+1}$ för $0 \leq i < n$ eller $l_i \geq l_{i+1}$ för $0 \leq i < n$.

14. (3p) Skriv en klass Exercise som kan registrera och presentera hur mycket tid du lägger på olika former av motion. Klassen ska ha två metoder:

- Med metoden `register` lägger man in motionsdata i form av en aktivitet och antal minuter man har lagt på aktiviteten.
- Metoden `show` skriver ut en sammanställning av din motion, med en enkel rubrik och sedan en rad per aktivitet och det totala antalet minuter registrerade på aktiviteten.

Din klass ska vara implementerad så att nedanstående exempel fungerar.

Exempelanvändning:

```
[In: ] workout = Exercise()
[In: ] workout.register('running', 20)
[In: ] workout.register('badminton', 60)
[In: ] workout.register('running', 30)
[In: ] workout.register('swimming', 25)
[In: ] workout.register('running', 15)
[In: ] workout.show()
[Out:]
Activity  Duration (min)
running         65
badminton       60
swimming        25
[In: ] homer = Exercise()
[In: ] homer.register('walked', 1)
[In: ] homer.show()
[Out:]
Activity  Duration (min)
walked           1
```

I exemplet är kolumnerna snyggt justerade, men det behöver du inte implementera.