

- 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`?

- 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!

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

2. Vad skrivs ut av koden till höger?

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

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

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

```
a = 1

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

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

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, \dots, x_n . Om listan är tom ska funktionen ge sårffallet `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 sårffall `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 $L = (l_0, l_1, \dots, l_n)$ 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.