

- 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 12 poäng).
- Inga import (Pythons standardbibliotek eller externa bibliotek) får användas om de inte nämns eller finns med i uppgiften. Man får 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. Vad innehåller `t` efter att man kört kodsnutten till höger?

- A. ''
- B. 'Lycka'
- C. 'LyckaTill'
- D. 'Till'
- E. 'LyckTil'

```
l = ["Lycka", "Till"]
t = ""
for i in range(len(l)):
    for x in l[i]:
        t = t + x
```

2. Vilken av följande kodrader läser in hela innehållet från en fil med namnet 'data.txt', efter att `f = open('data.txt', 'r')` exekverats?

- A. `all_contents = f.input()`
- B. `all_contents = f.readlines()`
- C. `all_contents = read(f)`
- D. `all_contents = flush(f)`
- E. `all_contents = f()`

3. Hur många gånger anropas `print` när man kör koden till höger?

- A. 1 gång
- B. 2 gånger
- C. 3 gånger
- D. 4 gånger
- E. 5 gånger

```
i = 0
j = 5
while i < 4 and j < 7:
    print(i, ",", j)
    i += 1
    j += 1
```

4. Hur många kombinationer av tilldelningar av True/False för variablerna x och y finns det som gör uttrycket (x or y) and x sant?

- A. 0
- B. 1
- C. 2
- D. 3
- E. 4

5. Vad händer när vi kör koden till höger?

- A. 7 skrivs ut
- B. 8 skrivs ut
- C. 10 skrivs ut
- D. Det uppstår ett särfall som indikerar att f inte kan ta två argument.
- E. Det uppstår ett särfall som indikerar att x förekommer både som global variabel och som parameter till f.

```
x = 2
def f(x,y=3):
    x += y
    return (x + y)
print(f(4, 2))
```

6. Vilket av följande anrop ger 5 givet m = {1:[1, 2, 3], 2:[4, 5, 6], 3:[7, 8, 9]} ?

- A. m.keys(2)[2]
- B. m[2][2]
- C. m[1][1]
- D. m[2][1]
- E. Det går ej att komma åt 5 från m utan att iterera.

7. Givet d = {'a':1, 'b':2, 'c':3, 'd':4}, vilket av följande uttryck skriver ut listan ['b', 'd']?

- A. `print([k for k in d if d[k] in [2,4]])`
- B. `print([d[2], d[4]])`
- C. `print([d['b'], d['d']])`
- D. `print(sorted(d.values())[1:4:2])`
- E. `print([k for k,v in d.items()])`
- F. `print(list(d))`

8. Vad blir resultatet av koden till höger?

- A. 1
- B. 2
- C. 3
- D. 4
- E. 5

```
def f(x):
    if x == 0:
        return x
    elif x % 2 == 0:
        return x - f(x-1)
    else:
        return x + f(x-1)
print(f(5))
```

Del B: kodfrågor

Var snäll använd ett papper till varje fråga i del B. Delfrågor får gärna lösas på samma papper.

9.

- A. Skriv en funktion `sum_vals(d)` som tar in en uppslagstabell `d` där alla värden är tal och returnerar summan av endast de jämna positiva heltalen. (2p)

Exempelanvändning:

```
[In: ] ex_dict1 = { 'a' : 2 , 'b' : 4, 'c' : 6 }
[In: ] ex_dict2 = { 'd' : -2 , 'e' : 4, 'f' : 3, 'g' : -1 }
[In: ] print(sum_vals(ex_dict1))
[Out:] 12
[In: ] print(sum_vals(ex_dict2))
[Out:] 4
```

- B. Skriv en funktion `max_value(d)` som tar in en uppslagstabell `d` där alla värden är tal och returnerar det största av dem. (1p)

Exempelanvändning (med samma `ex_dict1` och `ex_dict2` som ovan):

```
[In: ] print(max_value(ex_dict1))
[Out:] 6
[In: ] print(max_value(ex_dict2))
[Out:] 5
```

10.

- A. Skriv en funktion `read_numbers(h)` som läser in en textfil där varje rad innehåller antingen ett heltal eller en sträng. Parametern `h` är ett filhandtag, dvs filen har redan öppnats och förberedits för läsning. Funktionen ska:

- returnera en lista med alla heltal i filen,
- ignorera rader som inte går att tolka som heltal.

(2p)

Exempelanvändning:

```
## Ex: h pekar på filen data.txt som innehåller raderna:
## 10
## 3
## 99
## hej
## -5
[In: ] print(read_numbers("data.txt"))
[Out:] [10, 3, 99, -5]
```

- B. Skriv ytterligare en funktion `sum_file(filename)` som använder `read_numbers` (se uppgift 10A.) för att summera alla heltal i filen och returnerar summan. Funktionen ska returnera `False` om filen inte kan öppnas. (1p)

Exempelanvändning:

```
[In: ] print(sum_file("data.txt"))
[Out:] 107
```

11. Skriv en funktion `run_lengths(xs)` som tar en lista av heltal och returnerar en ny lista med längderna på varje följd av identiska tal, i ordning. (2p)

Exempelanvändning:

```
[In: ] print(run_lengths([1,1,1,2,2,3,1,1,4,4,4,4]))
[Out:] [3, 2, 1, 2, 4]
[In: ] print(run_lengths([1,9,3,7]))
[Out:] [1, 1, 1, 1]
[In: ] print(run_lengths([5,5,5,5,5,5]))
[Out:] [6]
[In: ] run_lengths([])
[Out:] []
```

12. Du arbetar på ett svenskt spelbolag som håller på att utveckla ett rollspel. Din uppgift är att skriva klasser som representerar spelkaraktärer och grupper av karaktärer.

- A. Skriv en klass `GameCharacter` som representerar en karaktär i spelet. Klassen ska ha en konstruktor som låter användaren sätta instansattributen `name`, `role` och `level`. Namn (`name`) och roll (`role`) representeras med strängar, och nivå (`level`) representeras som ett heltal. (1p)

Klassen ska även ha en `__str__`-metod som returnerar en sträng på formen

```
<name> (<role>) - level <level>
```

(se nedan för exempel).

(1p)

Exempelanvändning:

```
[In: ] c1 = GameCharacter("Alyra", "Mage", 12)
[In: ] c2 = GameCharacter("Brannik", "Warrior", 15)
[In: ] print(c1)
[Out:] Alyra (Mage) - level 12
[In: ] print(c2.role)
[Out:] Warrior
```

- B. Skriv en klass `Party` som representerar en grupp av karaktärer. Konstruktorn ska ta argumenten `name` (namnet på gruppen) och `members`, en lista av `GameCharacter`-objekt. Konstruktorn ska kontrollera att alla karaktärer har en *unik* roll (t.ex. får det inte finnas två *Mage* i samma grupp). Om inte alla roller är olika ska ett `ValueError` lyftas. (2p)

Exempelanvändning:

```
[In: ] g1 = Party("Dragon Slayers", [c1, c2])
[In: ] print(g1.name)
[Out:] Dragon Slayers

[In: ] c3 = GameCharacter("Selene", "Mage", 9)
[In: ] g2 = Party("Mystic Circle", [c1, c3])
[Out:] Traceback (most recent call last):
      ...
ValueError: Each role must be unique within the party
```