

- 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.
- **Skriv tydligt.** Svårlästa svar riskerar 0 poäng. Skriv bara på en sida av varje papper.
- Man måste bli godkänd på del A (4 rätt på 8 frågor) för att del B ska rättas.
- Del B består av ett antal frågor med varierande poäng (totalt 12).
- 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` om ej annat anges.
- 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. Hur kan man lagra ordet "Confundo" i variabeln `spell` i Python?

- A. `spell = [Confundo]`
- B. `spell = 'Confundo'`
- C. `spell = "Confundo"`
- D. `spell = #Confundo`
- E. `spell = Confundo`

2. Vilken eller vilka värden tolkas som "falskt" i en if-sats?

- A. `false`
- B. `False`
- C. `None`
- D. `""`
- E. `0`

3. Hur många trollformler skrivs ut av anropet `cast_spell(my_spells)` nedan?

- A. 0, koden är inte korrekt.
- B. 1
- C. 2
- D. 3
- E. 4

```
import random as r
my_spells = ['Alohomora', 'Ascendio',
             'Expelliarmus', 'Lumos']
def cast_spell(spells):
    i = 4
    while i > 1:
        print(r.choice(spells))
        i = i / 2
```

4. I funktionen `cast_spell(spells)` (ovan) används instruktionen `import random as r`. Vad är syftet med `as r`?

- A. Det ger mindre att skriva och ibland lite mer lättläst kod.
- B. Ordet `random` är ett nyckelord, så man måste döpa om modulen.
- C. Programmerare är skrockfulla.
- D. Det är för att förenkla särfallshantering.
- E. Det ger att slumpetal i variabeln `r` som vi senare använder.

5. Betrakta kodsnutten till höger. Vad blir resultatet av anropet `f([1,2,3,4])`?

- A. -8
- B. -2
- C. 0
- D. 2
- E. 8

```
def f(lst):  
    if lst == []:  
        return 0  
    else:  
        return lst[0] - f(lst[1:])
```

6. Vilken eller vilka tilldelningar gör att uttrycket `not (x and (y or z))` blir `True`?

- A. `x=False, y=False, z=False`
- B. `x=True, y=False, z=False`
- C. `x=False, y=True, z=False`
- D. `x=True, y=False, z=True`
- E. `x=False, y=False, z=True`

7. Vad blir det för utskrift av koden till höger?

- A. -15
- B. -5
- C. 6
- D. 7
- E. 15

```
def f(x=2, y=8):  
    y = 5  
    return x + y  
  
x = 10  
y = -15  
  
print(f())
```

8. Vad blir resultatet av `print([3 ** x for x in range(3)])`?

- A. `[0, 1, 2]`
- B. `[1, 3, 6]`
- C. `[1, 3, 6, 9]`
- D. `[1, 3, 9]`
- E. `[1, 3, 9, 27]`

Del B: kodfrågor

Var snäll använd ett papper (eller fler) till varje fråga i del B.

9. Skriv en funktion `common_max(list1, list2)` som tar in två listor och returnerar det största värdet som finns i båda. Om listorna inte har några gemensamma värden ska `0` returneras. (2p)

Obs: man får använda sig av den inbyggda funktionen `max` vilken returnerar maxvärdet i en lista.

Exempelanvändning:

```
[In: ] print(common_max([6,1,2,4],[2,3,5]))
[Out:] 2
[In: ] print(common_max([6,1,2,3,4],[4,2,3,6,7]))
[Out:] 6
[In: ] print(common_max([6,1,4],[2,3,5]))
[Out:] 0
```

10. Skriv funktionen `bar_diagram(lst)` som tar en lista med tal och skriver ut ett enkelt liggande stapeldiagram, med en stapel per tal i listan. Talen ska avrundas och eventuella negativa tal ska behandlas som talet `0`. Du kan utgå ifrån att alla element i listan är tal. (2p)

Obs: på den här uppgiften får man använda sig av den inbyggda funktionen `round` för att avrunda tal. Den funkar så att `round(x)` är det närmaste heltalet till `x`, så `round(1.3)` är `1` och `round(1.63)` är `2`, etc.

Exempelanvändning:

```
[In: ] bar_diagram([])
[Out:]
[In: ] bar_diagram([1,2,3,4])
[Out:]
#
##
###
####
[In: ] bar_diagram([10, 0, -1, 1, 0.1, 1.1, 0.9])
[Out:]
#####

#

#

#
```

11. Vi är vana vid att arbeta i bas 10 när vi räknar, i det decimala talsystemet, vilket betyder att vi använder siffrorna 0 till 9. I datorernas barndom var det praktiskt att arbeta i basen 16, *hexadecimalt*, och man la då till bokstäverna A till F för att representera värdena 10 (A) till 15 (F). Det hexadecimala talet FF betyder alltså "F gånger 16 plus F", vilket evalueras till $15 \cdot 16 + 15 = 255$. På samma sätt motsvarar det hexadecimala talet A2E talet 2606 i bas 10 ($10 \cdot 16^2 + 2 \cdot 16 + 14 = 2560 + 32 + 14 = 2606$).

Koden nedan innehåller en funktion som ska konvertera ett hexadecimalt tal, givet som en sträng, till decimala värden. Tyvärr finns det två fel, vilka? (2p)

```
def decode_hexadecimal(s):
    for c in s:
        if 'A' <= c and c <= 'F':
            val = 10 + ord(c) - ord('A')
        else:
            val = int(c)
        num = 16 * num + val
    return num
```

Obs: funktionen `ord` returnerar ett standardiserat tal som kodar bokstaven. Till exempel har "A" koden 65. Bokstäverna "A" till "F" ligger intill varandra i koden.

12. Skriv funktionen `pair_map(f, lst1, lst2)` som returnerar en lista med tripplar innehållande element från `lst1`, `lst2`, och funktionen `f` använd på de två elementen. Om indatalistorna har olika längd ska funktionen ignorera de sista elementen från den längre listan.

Du får **inte** använda den inbyggda funktionen `map` när du löser denna uppgift. **(2p)**

Exempelanvändning:

```
[In: ] pair_map(lambda x, y: x, [], [])
[Out:] []
[In: ] pair_map(lambda x, y: x+y, ['a'], ['b'])
[Out:] [('a', 'b', 'ab')]
[In: ] pair_map(lambda a, b: a+b, [1, 2, 3], [1, 1, 1])
[Out:] [(1, 1, 2), (2, 1, 3), (3, 1, 4)]
[In: ] pair_map(lambda a, b: a+b, [1, 2, 3], [10])
[Out:] [(1, 10, 11)]
[In: ] pair_map(lambda a, b: a*b, [1, -1, 1], [-2, 2, -2])
[Out:] [(1, -2, -2), (-1, 2, -2), (1, -2, -2)]
```

13. På den här uppgiften ska man skriva en klass för att representera rationella tal. All kod för deluppgifterna på den här uppgiften kan skrivas på samma papper.

- A. Skriv en klass `Rational` för att representera rationella tal (bråktalet) med en konstruktor som tar in två heltal `n` och `d`, och sätter instansattributen `numerator` till `n` och `denominator` till `d`. Om `d` är `0` så ska ett `ValueError` med lämpligt felmeddelande lyftas.¹ **(2p)**
- B. Lägg till en `__str__` metod som konverterar ett flyttal till en sträng på formen `n / d`. **(1p)**
- C. Lägg till en metod `normalize` som normaliserar flyttalet genom att dela täljare och nämnare med deras största gemensamma delare. **(1p)**

Obs: på den här uppgiften kan ni använda funktionen `gcd(x, y)` från `math` vilken beräknar största gemensamma delare av `x` och `y`.

Exempelanvändning:

```
[In: ] r = Rational(2,4)
[In: ] print(r)
[Out:] 2 / 4
[In: ] r.normalize()
[In: ] print(r)
[Out:] 1 / 2
[In: ] s = Rational(2,0)
[Out:] ValueError: cannot have denominator 0
```

¹*numerator* är täljare (talet ovan) och *denominator* är nämnaren (talet nedan) i ett bråktalet.