

- **Skriv tydligt.** Svårlästa svar riskerar 0 poäng.
 - Skriv bara på en sida av varje papper!
 - 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 (5 rätt på 10 frågor) för att del B ska rättas.
 - Ni behöver inte skriva dokumentationssträngar till funktioner och metoder.
 - Inga externa bibliotek får användas om de inte nämns eller finns med i uppgiften, man får dock använda inbyggda funktioner som `len` och `range`.
 - **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 svarpapper.

1. Vilka utav följande påståenden är korrekta baserat på information från kodraderna till höger?
A. Y är en klass.
B. Y ärver metoder från X.
C. X ärver metoder från Y.
D. Y ärver instansattribut från X.
E. X ärver instansattribut från Y.

```
class X(Y):  
    def __init__(self):  
        super().__init__()
```
2. Vad innebär begreppet *typkonvertering* (även känt som *type casting*)?
A. Man begränsar variabler till endast en typ. Till exempel kan man bestämma att `x` bara kan lagra heltalsvärden.
B. Man tolkar om ett värde till en annan typ. Till exempel kan ett heltal göras om till ett flyttal.
C. Man gör en avrundning.
D. Man använder en if-sats till att avgöra om ett värde är giltigt eller inte.
E. Man använder samma namn på metoder från olika klasser.
3. Hur många kombinationer av tilldelningar av `True/False` för variablerna `x`, `y`, och `z` finns det som gör uttrycket till höger sant?
A. 0
B. 1
C. 2
D. 3
E. 4

```
x and (y or z) and (not y or not z)
```
4. Om `x = [{'a': 1, 'b': 2}, 3, [4, 5], (6, 7)]`, vad ska man skriva för att komma åt värdet 2?
A. `x[0][1]`
B. `x[1][2]`
C. `x['b'][0]`
D. `x['b'][1]`
E. `x[0]['b']`

5. Vad blir värdet på variabeln `x` i koden till höger?

- A. 3
- B. 5
- C. 7
- D. 9

```
lis = [[1], [[1,2], [3,4], []], (6,9,2)]
x = sum([len(item) for item in lis])
```

E. Inget då ett särfall lyfts

6. Vad är syftet med `break`?

- A. Att omedelbart gå till början av en `for/while` loop.
- B. Att stänga fil för läsning/skrivning.
- C. Att avsluta en loop i förtid.
- D. Att orsaka radbrytning efter en `print`-sats.
- E. Inget, instruktionen finns inte i Python.

7. Vad skrivs ut av koden till höger?

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

```
a = 1
b = 2
def f(x):
    a = 3
    b = x + a
    return b

def g(a, b):
    x = f(a + b)
    return x

print(g(0, 1))
```

8. Om funktionen `fkn` är definierad som till höger, vad returnerar anropet `fkn(4)`?

- A. 2
- B. 3
- C. 4
- D. 5
- E. 6

```
def fkn(n):
    a = 0
    b = 1
    for i in range(n):
        c = a + b
        a = b
        b = c
    return b
```

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

- A. 0
- B. 4
- C. 8
- D. 10
- E. 11

```
l = [1,2,5,3]
x = 0
while len(l) > 1:
    x += l.pop()
print(x)
```

10. Vad har instruktionen `raise` för syfte?

- A. Utföra exponentiering av stora tal.
- B. Skapa alternativa identifierare när man importerar moduler.
- C. Lyfta ett särfall (*exception*) för att signalera problem.
- D. Höja prioriteten på en beräkning, så att andra beräkningar pausar.
- E. Det används i `print`-satser, för att till exempel kunna göra en fotnot (t.ex. "hubba¹").

Del B: kodfrågor

11. Här nedanför får du några kodrader givna i slumpmässig ordning och utan indentering. Ordna dem i rätt ordning och med rätt indentering så att koden ger en funktion som konverterar en sträng på

svenska till rövarspråket, dvs varje konsonant ersätts av samma konsonant två gånger med ett "o" emellan. Till exempel ska "koda" bli "kokododa" i rövarspråket. Funktionen antar att indata är ord skrivna med små bokstäver.

Kodraderna:

```
r += c
r += c + 'o' + c
r = ''
if c in 'aeiouyåäö':
else:
def sve2rovar(s):
for c in s:
return r
```

Exempelanvändning:

```
[In: ] sve2rovar('')
[Out:] ''
[In: ] sve2rovar('bo')
[Out:] 'bobo'
[In: ] sve2rovar('oklart')
[Out:] 'okoklolarortot'
```

12. Koden till höger är inspirerad av en annons i tunnelbanetågen för att rekrytera programmerare. Tanken är att koden ska skriva ut en "hemlig" sifferkombination. Om man försöker köra koden i Python finns det minst ett problem som orsakar ett exekveringsfel. Vilket?

```
a = '1112001984'

def hemlig_kod():
s = ''
for i in range(1, len(a)):
if (a[i] % 2) == (a[i-1] % 2):
s += max(a[i], a[i-1])
print(s)

hemlig_kod()
```

13. Skriv en funktion `div_by_five` som tar en lista med positiva heltal och returnerar en lista med alla element jämnt delbara med 5. Till exempel, `div_by_five([12, 5, 11, 30, 25])` ska returnera `[5, 30, 25]`.
14. Skriv en funktion `tabulate` som skriver ut nycklar och nycklarnas associerade värden i en uppslagstabell (eng: *dictionary*) så att man får en nyckel och dess värde per rad. Ex: Med uppslagstabellen

```
table = {'life': 42.0,
         'perfume': 47.11,
         'course': 'DA2004' }
```

ska `tabulate(table)` skriva ut

```
life 42.0
perfume 47.11
course DA2004
```

15. I uppgift 3 skulle du avgöra hur många tilldelningar som gör att uttrycket

```
x and (y or z) and (not y or not z)
```

evaluerar till `True`. Skriv kod för att lösa uppgiften!

16. Om man misstänker att det finns problem i en datamängd så kan man leta efter upprepade värden. Om det är "riktiga" mätvärden så är det ofta mycket osannolikt att man får samma avläsning mer än en gång. Skriv funktionen `repeated_numbers` som tar en lista med flyttal och returnerar en uppslagstabell (`dict`) som innehåller upprepade värden och antalet gånger de hittats i listan.

Exempel:

```
[In: ] repeated_numbers([])
[Out:] {}
[In: ] repeated_numbers([0.0, 0.0])
[Out:] {0.0: 2}
[In: ] repeated_numbers([0.0, 1.0, 0.0])
[Out:] {0.0: 2}
[In: ] repeated_numbers([0.0, 1.1, 2.2])
[Out:] {}
[In: ] repeated_numbers([0.0, 1.1, 0.0, 1.1, 2.2])
[Out:] {0.0: 2, 1.1: 2}
```

17. Skriv en rekursiv funktion `sum_rec(s)` som rekursivt summerar tal i en lista och returnerar summan. Till exempel, `sum_rec([2,5,1])` ska returnera 8. Obs: funktionen måste vara rekursiv för att man ska få poäng på uppgiften.

18. Betrakta funktionen nedan:

```
def f(lis):
    s = 0
    for x in lis:
        s += x**2
    return s
```

Skriv en ekvivalent version med endast `return`, `sum`, `map` och `lambda`. Dvs lösningen ska se ut på följande vis:

```
def f(lis):
    return <solution using sum, map and lambda here>
```

19. Skriv en klass `Card` som representerar ett kort (i en kortlek). Klassen ska endast ha en konstruktor som tar parametrarna `color` (heltal 1 till och med 4) och `rank` (heltal 1 till och med 13). Här är färg och värden representerade som heltal, med 1 för klöver, 2 för ruter, 3 för hjärter, och 4 för spader. Rangén ges av värden `[1,13]` där ess motsvarar 1 och kung 13. Om användaren skriver in ett ogiltigt tal ska ett `ValueError` lyftas.

Exempelanvändning:

```
[In] : c = Card(4,13) # possible colors: [1,4], possible values: [1,13]
[In] : print(c.color, c.rank)
[Out]: 4, 13
[In] : c = Card(2, 21)
[Out]: ValueError
```

20. Utöka klassen `Card` med metoden `__lt__`. Metoden `__lt__` ska jämföra värdet på två kort och returnera `True` eller `False`. Värdet på korten beräknas enligt följande. Ett kort med högre rang är mer värt än ett kort med lägre rang, oavsett färg. Vid samma rang på korten avgör färgen, där klöver < ruter < hjärter < spader. Dessutom ska vi anta att ess har högst rang, dvs $2 < 3 < 4 < \dots < \text{kung} < \text{ess}$. Spader ess har därför högst värde och klöver två lägst.

Exempelanvändning:

```
[In] : c1 = Card(4,13) # King of spades
[In] : print(c1)
[In] : c2 = Card(4,1) # Ace of spades
[In] : print(c1 < c2) # King lower value than Ace
[Out]: True
[In] : c3 = Card(2,13) # King of diamonds
[In] : print(c1 < c3) # Spades higher value than diamonds
[Out]: False
```