

- 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 (5 rätt på 10 frågor) för att del B ska rättas.
- **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 och samla svaren på del A på ett svarspapper. Varje fråga på del A är värd 1 poäng.

1. Betrakta kodsnutten till höger, vad blir värdet på `s` när man har kört koden?

A. 3

B. 8

C. 11

D. Inget då ett särfall lyfts p.g.a. att man inte kan ha listor i uppslagstabeller.

E. Inget då ett särfall lyfts då man inte kan beräkna längden på en uppslagstabell.

```
d = { 2 : "hej",  
      3 : [1,2,3],  
      4 : { 5 : "hej" , 6 : "du" } }
```

```
s = 0
```

```
for x in d:  
    s += len(d[x])
```

2. Vad skrivs ut om vi kör kodsnutten till höger?

A. Inget då vi inte anropat `f` med tillräckligt många argument.

B. 25

C. 7

D. Inget då `f` fastnar i en oändlig loop då vi skrivit `y = x + y`.

E. 6

```
x = 10
```

```
def f(x,y,z=3):  
    y = x + y  
    return (x + y + z)
```

```
print(f(1,2))
```

3. Betrakta listomfattningen `[x * 2 for x in range(0,10) if x % 2 == 0]`, vilken av kodsnutterna nedan producerar samma lista?

A. `list(map(lambda x: x * 2, reduce(lambda x: x % 2 == 0, range(0,10))))`

B. `list(filter(lambda x: x % 2 == 0, map(lambda x: x * 2, range(0,10))))`

C. `list(map(lambda x: x * 2, range(0,10)))`

D. `list(map(lambda x: x * 2, filter(lambda x: x % 2 == 0, range(0,10))))`

E. Inget av alternativen ovan, man måste använda en `for` eller `while` loop för att producera listan.

4. Vilken/vilka av följande tilldelningar gör att `x or (not y and z)` blir `True`?

A. `x = True, y = False, och z = True`

B. `x = False, y = False, och z = True`

C. `x = True, y = True, och z = True`

- D. `x = False, y = False, och z = False`
- E. `x = False, y = True, och z = True`

5. Betrakta kodsnutten till höger, vad blir resultatet av `f([9, 2, 1, 3])`?

- A. 15
- B. 6
- C. 10
- D. 4
- E. 3

```
def f(mylist):  
    if mylist == []:  
        return 0  
    else:  
        return len(mylist) + f(mylist[1:])
```

6. Vilka av följande påståenden är sanna om Python?

- A. Med termen *högre ordningens funktioner* menas funktioner som kan hantera godtyckligt stora tal.
- B. Subklasser kan bara ha en superklass.
- C. Strängar är muterbara.
- D. Listor är muterbara.
- E. `range` returnerar ett "`range`-objekt", inte en lista.

7. Med vad skapar man ett särfall (eng. exception) i Python?

- A. `try`
- B. `except`
- C. `throw`
- D. `raise`
- E. `exception`

8. Betrakta kodsnutten till höger, vad händer om vi kör `divide(2, 0)`?

- A. Ett `AssertionError` lyfts.
- B. Talet 2 returneras.
- C. Ett `ZeroDivisionError` lyfts.
- D. Talet 0 returneras.
- E. Först skrivs `AssertionError: assert y != 0` ut och sen lyfts ett `ZeroDivisionError`.

```
def divide(x, y):  
    assert y != 0  
    return x / y
```

9. Vad händer om man kör kodsnutten nedan till höger?

- A. Ingenting då vi direkt fastnar i en oändlig loop.
- B. Användaren får skriva in tal tills hen skrivit in 42 och då avslutas loopen och `Congratulations!` skrivs ut.
- C. Inget då det inte går att skriva en `while` loop med en variabel på detta sätt utan man måste ha ett booleskt test.
- D. Användaren får skriva in tal, men loopen avslutas aldrig då `x` inte konverteras till ett heltal innan det jämförs med 42.
- E. Användaren får skriva in tal och varje gång hen skriver 42 skrivs `Congratulations!` ut, men loopen avslutas aldrig då vi inte har med något `break`.

```
b = True  
  
while b:  
    x = input("Write a number: ")  
  
    if x == 42:  
        print("Congratulations!")  
        b = False
```

10. Betrakta kodsnutten till höger, vad är värdet på `out` när den har körts?

- A. Ppyttthhhhooonnnnn
- B. Python
- C. nnnnnnoooooohhhttttyP
- D. nnnnnnoohhhtty
- E. ytthhhooonnnnn

```
s = "Python"
out = ""

for i in range(0, len(s)):
    out += i * s[i]
```

Del B: koduppgifter

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

11. Nedan följer ett försök att skriva ett program för att beräkna alla tal mindre än `v` som delar `v`:

```
v = 42
divs = []

for x in range(0, v):
    if v % x == 0:
        divs.append(x)
```

Koden innehåller dock en bugg och det lyfts ett särfall om man försöker köra den. Uppgiften är nu att modifiera koden så att den fungerar som den ska samt förbättra den så att man istället har en funktion `divisors(v)` som tar in `v` som argument istället för en global variabel. (1p)

Exempelanvändning:

```
[In: ] print(divisors(42))
[Out:] [1, 2, 3, 6, 7, 14, 21]
[In: ] print(divisors(62))
[Out:] [1, 2, 31]
```

12. Skriv en funktion `exponents(x, mylist)` med hjälp av `map` och `lambda` som tar in ett tal `x` och en lista av tal `mylist` och sedan returnerar en lista med `x` upphöjt till varje tal i listan. (1p)

Exempelanvändning:

```
[In: ] print(exponents(2, [1, 2, 3]))
[Out:] [2, 4, 8]
[In: ] print(exponents(3, [5, 2, 1, 3]))
[Out:] [243, 9, 3, 27]
```

Obs: för poäng måste `exponents` vara skriven med `map` och `lambda`.

13. Skriv en funktion `word_length(f)` som tar in ett filnamn `f` och skriver ut längden på varje ord i filen med ett mellanslag mellan varje ordlängd samt radbrytningar på samma platser som i filen. (1p)

Exempelanvändning: givet en fil `indata.txt` som innehåller

```
Hej hej,
lycka till på tentan!
```

så ska `word_length('indata.txt')` skriva ut:

```
3 4
5 4 2 7
```

Obs: specialtecken som `,` `'` och `!` ska hanteras som vilka bokstäver som helst.

14. ASCII är en standard för att representera bokstäver som tal. Med denna standard är t.ex. "a" representerad som 97, "b" representerad som 98, etc., ända upp till "z" som är representerad som 122. Python har en funktion `chr` som returnerar en bokstav givet ett tal enligt ASCII standarden, så `chr(97)` är a, `chr(98)` är b, etc.

Skriv en kodsnuett som använder `chr` för att skapa en uppslagstabell `letters` från små bokstäver till deras position i alfabetet räknat från 1 (så nyckeln 'a' ska ha värde 1, nyckeln 'b' ska ha värde 2, etc.). (1p)

Obs: för poäng får man inte hårdkoda uppslagstabellen utan den ska skapas genom att `chr` funktionen används på ett lämpligt sätt i exempelvis en loop eller dict-comprehension.

15. Skriv en funktion `sum_string(s,d)` som tar in en sträng `s` samt en uppslagstabell `d` från bokstäver till tal och returnerar summan av de tal som bokstäverna i `s` pekar på i `d`. (1p)

Exempelanvändning med uppslagstabellen `letters` som definierats i uppgift 14 ovan:

```
[In: ] print(sum_string("hej", letters))
[Out:] 23
```

Vi får detta resultat då "h" är bokstav 8, "e" bokstav 5 och "j" bokstav 10.

16. Skriv en *rekursiv* funktion `even_odd_sum(mylist)` som givet en lista `mylist` med tal beräknar en summa av talen på ett sånt sätt att jämna tal adderas medans udda tal subtraheras. (1p)

Exempelanvändning:

```
[In: ] print(even_odd_sum([1,2,3]))
[Out:] -2
[In: ] print(even_odd_sum([2,4,6,8]))
[Out:] 20
[In: ] print(even_odd_sum([1,3,6,3]))
[Out:] -1
```

Obs: för poäng måste funktionen vara rekursiv.

17. I Sagan om ringen böckerna är orcher (eng. orc) en typ av monster. Skriv en klass `Orc` så att när man anropar konstruktorn sätts *instansattributen* `hp` och `strength` med värden som ges av användaren. Dessa attribut representerar orchens liv och styrka. Skriv även en metod `beats` som givet en annan orch kollar vem som är starkast och returnerar `True` om orchen som man anropar metoden från är lika stark eller starkare än den andra orchen och `False` annars. (1p)

Exempelanvändning:

```
[In: ] orc1 = Orc(10,20)           # Orc with 10 HP and 20 strength
[In: ] orc2 = Orc(8,10)           # Orc with 8 HP and 10 strength
[In: ] print(orc1.beats(orc2))    # orc1 beats orc2 as it has higher strength
[Out:] True
```

18. En speciell typ av orcher är Uruk-hai. Dessa lyder under trollkarlen Saruman och är extra starka och kraftfulla. Skriv en klass `UrukHai` som ärver från `Orc` och har *klassattributen* `master` satt till 'Saruman' samt en konstruktör som använder sig av konstruktorn i superklassen `Orc`, men dubblerar både den `hp` och `strength` som man skapar Uruk-haien med. (1p)

Exempelanvändning:

```
[In: ] urukhai = UrukHai(15,15)   # Uruk-hai with 2*15 HP and 2*15 strength
[In: ] print(urukhai.beats(orc1)) # The Uruk-hai beats the stronger of the two orcs
[Out:] True
[In: ] print(orc2.beats(urukhai)) # so the weaker orc also doesn't beat the Uruk-hai
[Out:] False
[In: ] print(urukhai.master)     # The master of all Uruk-hai is Saruman
[Out:] Saruman
```

19. Skriv ett program som frågar användaren vad produkten av två slumpmässigt valda tal mellan 0 och 10 är. Om användaren skriver in korrekt produkt ska programmet gratulera hen och börja om med två nya tal. Om användaren istället skriver in fel produkt ska programmet upplysa användaren om detta samt skriva ut vad

den korrekta produkten är och sen börja om. Om användaren inte skriver något (dvs tomma strängen '') så ska programmet avslutas. (1p)

Vidare, om användaren skriver in något som inte är ett tal eller tomma strängen ska inte programmet krascha utan eventuellt fel ska hanteras på ett lämpligt sätt med hjälp av `try-except`. (1p)

Exempelkörning:

```
What is 8 * 9? 78
Incorrect! The correct answer is 72
What is 9 * 7? 63
Correct!
What is 2 * 5? apabepa
Input is not a number! The correct answer is 10
What is 1 * 0? 0
Correct!
What is 7 * 7?
Goodbye! The correct answer was 49
```

Användaren har i körningen ovan skrivit in 78, 63, apabepa, 0, och slutligen ingenting så därför avslutades programmet.

Man får använda funktionen `randint` från `random` biblioteket på den här uppgiften. Den fungerar på så sätt att `randint(0,10)` är ett slumpmässigt tal mellan 0 och 10.