

- 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 samla svaren på del A på ett svarspapper.

1. Vilka av följande tilldelningar gör formeln x **and** (**not** y **or** z) sann?

- A. $x = \text{True}, y = \text{True}, z = \text{True}$
- B. $x = \text{False}, y = \text{True}, z = \text{True}$
- C. $x = \text{True}, y = \text{False}, z = \text{True}$
- D. $x = \text{True}, y = \text{True}, z = \text{False}$
- E. $x = \text{True}, y = \text{False}, z = \text{False}$

2. Vad gäller för Python 3?

- A. Listor är omuterbara
- B. Strängar är omuterbara
- C. En klass kan ärva från flera klasser
- D. Typen på resultatet av **input** beror på vad användaren skriver in
- E. Iteration kan endast göras med **for** och **while**

3. Vilka av följande alternativ är typer i Python 3?

- A. float
- B. char
- C. module
- D. dict
- E. yields

4. Om s är en sträng och x en bokstav så returnerar $s.count(x)$ antal förekomster av x i s . Givet $s = \text{'DA2004'}$, vad blir då resultatet av $\{x : s.count(x) \text{ for } x \text{ in } s\}$?

- A. $\{1 : \text{'D'}, 1 : \text{'A'}, 1 : \text{'2'}, 2 : \text{'0'}, 1 : \text{'4'}\}$
- B. $\{(\text{'D'}, 1), (\text{'A'}, 1), (\text{'2'}, 1), (\text{'0'}, 2), (\text{'4'}, 1)\}$
- C. $[(\text{'D'}, 1), (\text{'A'}, 1), (\text{'2'}, 1), (\text{'0'}, 2), (\text{'4'}, 1)]$
- D. $\{\{\text{'D'}: 1\}, \{\text{'A'}: 1\}, \{\text{'2'}: 1\}, \{\text{'0'}: 2\}, \{\text{'4'}: 1\}\}$
- E. $\{\text{'D'}: 1, \text{'A'}: 1, \text{'2'}: 1, \text{'0'}: 2, \text{'4'}: 1\}$

5. Vad blir värdet på `out` efter att man kört koden nedan till höger?

- A. xxxxxx
- B. xx2004
- C. xx22004444
- D. xx224444
- E. 444422xx

```
s = "DA2004"
out = ""
for x in s:
    try:
        for y in range(int(x)):
            out += x
    except:
        out += "X"
```

6. Vad skrivs ut om man kör koden nedan till höger?

- A. Först 100 och sen 42
- B. Först 42 och sen -10
- C. Först -10 och sen 42
- D. Först -10 och sen -10
- E. Först 100 och sen -10

```
x = 42

def f(x = -10):
    print(x)

f(100)
f()
```

7. Vilka av följande påstående gäller för koden nedan till höger?

- A. `lista` är tom efter att man kört kodsnutten
- B. 0 skrivs ut
- C. `[4, 3, 1, 2, 5, 0]` skrivs ut
- D. `[4, 5]` skrivs ut
- E. 5 skrivs ut

```
def foo(mylist):
    m = 0
    while mylist:
        x = mylist.pop()
        if x > m:
            m = x
    return m

lista = [4, 3, 1, 2, 5, 0]
print(foo(lista))
```

8. Betrakta listomfattningen `[list(range(x)) for x in range(5)]` vilken av kodsnuttarna nedan producerar samma lista?

- A. `list(map(lambda x: list(range(x)), range(5)))`
- B. `list(filter(lambda x: list(range(x)), range(5)))`
- C. `list(map(lambda x: list(range(5)), range(x)))`
- D. `list(map(list, map(range, range(5))))`
- E. Inget av alternativen ovan då man inte kan ha en `range` som resultat i en listomfattning.

9. Givet `mylist = [1, 2, 3, 4]`, vilket eller vilka av följande `print`-uttryck skriver ut listan `[1, 3]`?

- A. `print([x - 1 for x in mylist if x % 2 == 0])`
- B. `print(mylist[0::2])`
- C. `print([x for x in mylist if x % 2 != 0])`
- D. `print([mylist[-4], mylist[-2]])`
- E. `print([mylist[0]] + [mylist[2]])`

10. Om en fil börjar med `from mod import fun as f` hur använder man funktionen `fun` i modulen `mod`?

- A. `fun`
- B. `f`
- C. `mod.fun`
- D. `mod.f`
- E. `fun.f`

Del B: kodfrågor

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

11. På ordinarie tenta 2023-01-12 skulle man skriva en funktion `float_to_percent(x)` som tog in ett flyttal `x` och returnerade en sträng med vad `x` är i procent. Skriv nu `percent_to_float(s)` som går åt andra hållet, d.v.s. givet en sträng `s` med ett procenttal ska ett flyttal som motsvarar procentvärdet returneras. (1p)

Exempelanvändning:

```
[In: ] print(percent_to_float("1%"))
[Out:] 0.01
[In: ] print(percent_to_float("120%"))
[Out:] 1.2
[In: ] print(percent_to_float("0%"))
[Out:] 0.0
```

12. Deluppgifterna på den här uppgiften bör skrivas på samma papper.

- A. Skriv en funktion `reverse_word(s)` som tar in en sträng `s` och returnerar den baklänges. (1p)

Obs: man får inte använda funktionen `reversed` på den här uppgiften.

Exempelanvändning:

```
[In: ] print(reverse_word("DA2004"))
[Out:] 4002AD
[In: ] print(reverse_word("Tentan"))
[Out:] natneT
```

- B. Skriv en funktion `reverse_words(s)` som tar in en sträng `s` och returnerar en sträng med varje ord bakvänt. (1p)

Obs: man får använda `reverse_word` här även om man inte löst den uppgiften.

Exempelanvändning:

```
[In: ] print(reverse_words("Lycka till på tentan"))
[Out:] akcyL llit åp natnet
```

13. Skriv en *rekursiv* funktion `pair_with(x, mylist)` som tar in ett värde `x` samt en lista `mylist` och returnerar en lista där alla värden i `mylist` parats ihop i en tupel där `x` är första värdet. (1p)

Obs: för poäng måste `pair_with` vara rekursiv (d.v.s. anropa sig själv på ett lämpligt sätt).

Exempelanvändning:

```
[In: ] print(pair_with(2, [1, 2, 3]))
[Out:] [(2, 1), (2, 2), (2, 3)]
```

14. Allspråket är ett kodspråk där alla konsonanterna före första vokalen flyttas till slutet av ordet och "all" sedan läggs till i slutet, t.ex. "knasig" blir alltså "asigknall" (d.v.s. "asig-kn-all"). Skriv en funktion `till_allspraket(word)` som tar in en sträng `word` på svenska och översätter det till allspråket. För enkelhets skull kan man anta att `word` bara innehåller gemener (d.v.s. små bokstäver) och inga specialtecken. (2p)

Tips: vokalerna i svenska är aeiouyåäö.

Exempelanvändning:

```
[In: ] print(till_allspraket("knasig"))
[Out:] asigknall
[In: ] print(till_allspraket("omtenta"))
[Out:] omtentaall
```

15. Betrakta koden nedan som ber användaren om ett tal x mellan 1 och 10, sen försöker öppna filen "sample x .csv" (där x är talet användaren skrev in) och sen anropar funktionen `do_something` med filhandtaget.

```
while True:
    x = input("Which sample file do you want to open? ")

    if 1 <= int(x) and int(x) <= 10:
        h = open("sample" + x + ".csv")
        do_something(h)
        h.close()
        break
```

Koden har ett antal problem och uppgiften är att fixa till den enligt:

- Skriv om koden så att den använder **with** istället för **open-close**. (1p)
- Skriv om koden så att den har lämplig felhantering med hjälp av **try-except**. De fel som ska hanteras är:
 - (a) Användaren skriver inte in ett tal och konverteringen `int(x)` går därför inte att göra. (0.5p)
 - (b) Filen som man försöker öppna finns inte. (0.5p)Dessa två fel ska hanteras separat så att användaren får ett lämpligt meddelande i vardera fall.

Lösningen kan skrivas på en sida och man behöver bara lämna in en kodsutt som gör en eller två av de saker som efterfrågas.

16. På den här uppgiften ska man skriva klasser för att representera matriser. All kod för deluppgifterna på den här uppgiften kan skrivas på samma papper.

- A. Skriv en klass `Matrix` för att representera matriser med en konstruktor som tar in en lista av lika långa listor `m` och sätter instansattributen `matrix` till `m`. Konstruktorn ska även sätta instansattributen `rows` och `cols` till antalet rader respektive kolumner i matrisen. (1p)

Obs: man kan anta att indata är en icke-tom lista av lika långa icke-tomma listor.

Exempelanvändning:

```
[In: ] m1 = Matrix([[1,2,3],[4,5,6]])
[In: ] print(m1.matrix)
[Out:] [[1, 2, 3], [4, 5, 6]]
[In: ] print(m1.rows)
[Out:] 2
[In: ] print(m1.cols)
[Out:] 3
```

- B. Lägg till subklassen `Identity` som ärver från `Matrix` och representerar enhetsmatriser. Konstruktorn ska ta in ett tal n och anropa konstruktorn i `Matrix` med en enhetsmatris av storlek $n \times n$ (d.v.s. en $n \times n$ matris med ettor på diagonalen och nollor på alla andra positioner). (1p)

Exempelanvändning:

```
[In: ] id3 = Identity(3)
[In: ] print(id3.matrix)
[Out:] [[1, 0, 0], [0, 1, 0], [0, 0, 1]]
[In: ] print(id3.rows)
[Out:] 3
[In: ] print(id3.cols)
[Out:] 3
```