

1. Seuraavassa esitetään neljä Python-ohjelmaa. Kirjoita kunkin ohjelman viereen sen tuottama tuloste (kukin kohta 3 p., yht. 12 p.).

Ohjelma 1: / Program 1: <pre>n = 100 output = "X" if n >= 10: print(output) if n > 150: output = "Y" else: output = "Z" else: print("E") print(output)</pre>	Tuloste / Output:
Ohjelma 2: / Program 2: <pre>i = 1 while i % 5 != 0: print(i) i = i + 2 print(i)</pre>	
Ohjelma 3: / Program 3: <pre>for num in range(6, 1, -2): print(f"{num}. item")</pre>	
Ohjelma 4: / Program 4: <pre>people = ("Maria", "Ahmed", "Donald", "Olga") p1, p2, p3, p4 = people print(p2) print(people[2]) if p1 not in people: print(people)</pre>	

2. Kirjoita alla olevaan ohjelmaan puuttuvat kaksi funktiota siten, että ohjelma toimii tulostelaatikossa esitetyn esimerkin mukaisesti.

Ohjelma / Program:

Koodaa tähän / Code here:

```
# ---
# Tulostetaan käyttäjän antaman arvo 3-kertaisena.
number = int(input("Syötä kokonaisluku / Enter an integer: "))
result = calculate(number)
print(f"Lukusi 3-kertaisena / Your integer tripled: {result}")

# Tulostetaan niiden listassa olevien lukujen summa, jotka ovat sallittujen
# arvojen (nyt 5...15, päätepisteet mukaan lukien) sisällä. Ohjelman täytyy toimia oikein myös muilla
# sallituilla arvoilla.
list = [15, 1, 8, 18, 33]
borders = (5, 15)
result = operate(list, borders)
print(f"Sallittujen lukujen summa / sum of tolerated integers: {result}")
```

Tuloste / Output:

```
Syötä kokonaisluku / Enter an integer: 4
Lukusi 3-kertaisena / Your integer tripled: 12
Sallittujen lukujen summa / sum of tolerated integers: 23
```

3. Seuraavassa on Python-ohjelma, joka käyttää relaatiotietokantaa onnistuneesti. Ohjelman alapuolella on yhdeksän väitettä. Ota kunkin väitteen osalta kantaa siihen, onko väite tosi vai epätosi. (6 oikein = 1 p., 7 oikein = 2 p., 8 oikein = 3 p., 9 oikein = 4 p.).

```
import mysql.connector

1 usage

def fetchEmployees():
    sql = "SELECT id, lastname, firstname, salary FROM person"
    dbcursor = dbconn.cursor()
    dbcursor.execute(sql)
    result = dbcursor.fetchall()
    if dbcursor.rowcount>0:
        for row in result:
            print(f"Salary of {row[2]} {row[1]}: {row[3]} euros per month.")
    return

dbconn = mysql.connector.connect(
    host='127.0.0.1',
    port=3306,
    database='jengi',
    user='apps23',
    password='sAlasana_',
    autocommit=True
)

fetchEmployees()
```

Oikein	Väärin	Väite
		Ohjelman suoritus saa tietokantapalvelinohjelmiston käynnistymään.
		Tietokantapalvelinohjelmisto sijaitsee samassa tietokoneessa kuin Python-ohjelma.
		Ohjelma muodostaa täsmälleen yhden tietokantayhteyden.
		Ohjelma tulostaa toisen, ensimmäisen ja kolmannen henkilön palkan (salary) tässä järjestyksessä.
		Ohjelma saa hakea tietokannasta korkeintaan 3306 tietuetta.
		Ohjelman ajo saa tietokantapalvelimen suorittamaan SELECT -lauseen yhden kerran.
		Ohjelman tuottamasta tulosteesta nähdään henkilöiden tunnisteet (id).
		Ohjelma päivittää tietokannassa olevia henkilöiden tunnisteita (id).
		Tietokantataulu nimeltä person ei saa olla tyhjä.