

Certainly, let's get started on your tasks!

Task 1:

English: How can you write a nested conditional statement in Python? (1 point)

Armenian: Ինչպե՞ս կարելի է գրել բնադրված պայմանական արտահայտություն Python-ում: (1 միավոր)

Solution:

Բնադրված պայմանական արտահայտությունները Python-ում օգտագործվում են, երբ մեկ պայմանական արտահայտության ներսում գտնվում է մեկ այլ պայմանական արտահայտություն: Սա թույլ է տալիս ստուգել բազմակի պայմաններ և կատարել տարբեր գործողություններ, կախված այդ պայմանների ճշմարտության արժեքից:

Օրինակ:

```
x = 10
```

```
y = 5
```

```
if x > 0:
    if y > 0:
        print("Both x and y are positive")
    else:
        print("x is positive, but y is not")
else:
    print("x is not positive")
```

Task 2:

English: How can you check the data type of a variable in Python? (1 point)

Armenian: Ինչպե՞ս կարելի է ստուգել փոփոխականի տվյալների տիպը Python-ում: (1 միավոր)

Solution:

type() ֆունկցիան օգտագործվում է փոփոխականի տվյալների տիպը ստուգելու համար:

Օրինակ:

```
x = 10
```

```
y = "Hello"
```

```
z = 3.14
```

```
print(type(x))    # Output: <class 'int'>
print(type(y))    # Output: <class 'str'>
print(type(z))    # Output: <class 'float'>
```

Task 3:

English: How do you convert one data type to another in Python? Give an example. (1 point)

Armenian: Ինչպե՞ս կարելի է մեկ տվյալների տիպը վերածել մյուսի Python-ում:

Տրամադրեք օրինակ: (1 միավոր)

Solution:

Python-ում տվյալների տիպերի փոխակերպումը կատարվում է տարբեր ֆունկցիաների միջոցով:

- **int():** վերածում է թվային տող կամ լողացող թվի ամբողջ թվի:
- **float():** վերածում է թվային տող կամ ամբողջ թվի լողացող թվի:
- **str():** վերածում է ցանկացած տիպի օբյեկտը տողի:

Օրինակ:

```

x = "10"
y = 3.14

x_int = int(x)  # x_int is now an integer 10
y_int = int(y)  # y_int is now an integer 3

print(type(x_int))  # Output: <class 'int'>
print(type(y_int))  # Output: <class 'int'>

```

Task 4:

English: What is Python, and what are its main features? (1 point)

Armenian: Ի՞նչ է Python-ը և ինչեր են նրա հիմնական հատկությունները: (1 միավոր)

Solution:

Python-ը բարձր մակարդակի, ընթերցելի և բազմագործառույթ ծրագրավորման լեզու է: Այն հայտնի է իր հետևյալ հատկություններով:

- **Einfachheit:** Python-ի սինտաքսիսը պարզ և հասկանալի է:
- **Vielseitigkeit:** Python-ը կարող է օգտագործվել տարբեր ոլորտներում, ինչպիսիք են տվյալների գիտությունը, արհեստական բանականությունը, վեբ զարգացումը և այլն:
- **Große Standardbibliothek:** Python-ը ունի ընդարձակ ստանդարտ գրադարան, որը ապահովում է տարբեր գործառույթներ, ինչպիսիք են ֆայլային I/O, ցանցային ծրագրավորում, թվաբանական գործողություններ և այլն:
- **Plattformunabhängigkeit:** Python ծրագրերը կարող են գործարկվել տարբեր օպերացիոն համակարգերում:

Task 5:

English: Write a Python function `is_leap` that checks if a given year is a leap year or not and return True or False accordingly. (4 points)

Armenian: Գրեք Python ֆունկցիա `is_leap`, որը ստուգում է, թե տրված տարին նահատակային է, թե ոչ, և վերադարձնում է True կամ False համապատասխանաբար: (4 միավոր)

Solution:

```

def is_leap(year):
    if year % 4 == 0:
        if year % 100 == 0:
            if year % 400 == 0:
                return True
            else:
                return False
        else:
            return True
    else:
        return False

```

Task 6:

English: Write a function that squares each number in the given list and prints the resulting list. (4 points)

Armenian: Գրեք ֆունկցիա, որը քառակուսացնում է տրված ցուցակի յուրաքանչյուր թիվը և տպում արդյունքում ստացված ցուցակը: (4 միավոր)

Solution:

```
def square_numbers(numbers):  
    squared_numbers = [num**2 for num in numbers]  
    print(squared_numbers)
```

Task 7:

English: Write a function that takes two strings as input and returns True if they are anagrams of each other and False otherwise. Two strings are anagrams if they contain the same characters in the same frequency, but possibly in a different order. (8 points)

Armenian: Գրեք ֆունկցիա, որը վերցնում է երկու տող որպես մուտքային տվյալներ և վերադարձնում է True, եթե դրանք միմյանց անագրամներ են, և False հակառակ դեպքում: Երկու տողեր անագրամներ են, եթե դրանք պարունակում են նույն նշանները նույն հաճախականությամբ, բայց հնարավոր է տարբեր հերթականությամբ: (8 միավոր)

Solution:

```
def is_anagram(str1, str2):  
    if len(str1) != len(str2):  
        return False  
  
    char_count = {}  
    for char in str1:  
        char_count[char] = char_count.get(char, 0) + 1  
  
    for char in str2:  
        if char not in char_count or char_count[char] == 0:  
            return False  
        char_count[char] -= 1  
  
    return True
```

Let me know if you have any other tasks or questions!

- <https://medium.com/@abelkrw/a-beginners-guide-to-getting-started-with-python-ffe0d788f1a2>
- <https://github.com/lydiecherilus/python-exercises>
- https://github.com/Gheetha76/Internship_project
- <https://github.com/Piusin/Google-Africa-DEB-main>