

Here are the translations of the Python questions from the image, along with their solutions in Armenian:

Question 1: Ինչպե՞ս է գործում Python-ի ինտերպրետատորը: (1 միավոր)

Answer: Python-ի ինտերպրետատորը կարդում է և կատարում է Python կոդը տող առ տող: Այն վերլուծում է յուրաքանչյուր տող, թարգմանում է այն համակարգչի հասկանալի հրամանների և կատարում է այն:

Question 2: Բացատրե՛ք հավաքածուների (sets) հասկացությունը Python-ում: (1 միավոր)

Answer: Հավաքածուները (sets) Python-ում ներկայացնում են անկարգված և կրկնվող տարրեր չպարունակող տվյալների կառուցվածք:

Question 3: Ի՞նչ են տրամաբանական օպերատորները Python-ում և ինչպե՞ս են գործում: (1 միավոր)

Answer: Python-ում տրամաբանական օպերատորներն են and, or և not:

- **and:** Եթե երկու օպերանդներն էլ ճշմարիտ են, ապա արտադրանքը ճշմարիտ է:
- **or:** Եթե գոնե մեկ օպերանդը ճշմարիտ է, ապա արտադրանքը ճշմարիտ է:
- **not:** Փոխում է օպերանդի արժեքը՝ ճշմարիտը դարձնելով սուտ և հակառակը:

Question 4: Ի՞նչ է մեկնաբանությունների նշանակությունը Python կոդում: (1 միավոր)

Answer: Մեկնաբանությունները օգտագործվում են կոդը ավելի ընթերցելի և հասկանալի դարձնելու համար: Դրանք չեն ազդում կոդի կատարման վրա: Python-ում մեկնաբանությունները սկսվում են # նշանով:

Question 5: Խնդրե՛ք օգտատիրոջը մուտքագրել տող: Տպե՛ք տողում բառերի քանակը, ենթադրելով, որ բառերը բաժանված են տարածություններով: (4 միավոր)

```
text = input("Մուտքագրե՛ք տող: ")
words = text.split()
print("Բառերի քանակը:", len(words))
```

Question 6: Գրե՛ք Python ֆունկցիա, որը ընդունում է ամբողջ թվերի ցուցակ և վերադարձնում է նոր ցուցակ, որը պարունակում է սկզբնական ցուցակի բոլոր զույգ թվերի քառակուսիները, օգտագործելով ցուցակային ըմբռնում: (4 միավոր)

```
def squares_of_even_numbers(numbers):
    return [x**2 for x in numbers if x % 2 == 0]
```

```
# Example usage:
numbers = [1, 2, 3, 4, 5, 6]
result = squares_of_even_numbers(numbers)
print(result)  # Output: [4, 16, 36]
```

Question 7: Գրե՛ք ֆունկցիա, որը գեներացնում է Ֆիբոնաչի հաջորդականությունը մինչև n-րդ տերմինը և վերադարձնում է այն որպես ցուցակ: (8 միավոր)

```
def fibonacci(n):
    if n <= 0:
        return []
    elif n == 1:
        return [0]
    else:
        fib_list = [0, 1]
        for i in range(2, n):
            fib_list.append(fib_list[i-1] + fib_list[i-2])
        return fib_list
```

```
# Example usage:  
n = 10  
result = fibonacci(n)  
print(result)  # Output: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

Please let me know if you have any other questions or tasks!

- https://github.com/munga06/alx_python