

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 գործում է հետևյալ կերպ. նա կարդում է Python կոդը տող առ տող, վերլուծում է այն և կատարում է համապատասխան գործողությունները:

**Question 2:** 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 արտադրվում է ճշմարիտ է:
- **or:** Եթե գոնե մեկ օպերանդը ճշմարիտ է, ապա արտադրվում է ճշմարիտ է:
- **not:** Փոխում է օպերանդի ճշմարտության արժեքը:

**Question 3:** Ի՞նչ նշանակություն ունեն մեկնաբանությունները Python կոդում: (1 միավոր)

**Answer:** Մեկնաբանությունները օգտագործվում են կոդը հասկանալի դարձնելու համար: Դրանք չեն կատարվում ինտերպրետատորի կողմից:

**Question 4:** Ի՞նչ է փոփոխականի գործունակության շրջանակը և ինչպե՞ս է գործում Python-ում: (1 միավոր)

**Answer:** Փոփոխականի գործունակության շրջանակը սահմանում է, թե որտեղից կարելի է հասնել փոփոխականին: Python-ում կան երկու տեսակի գործունակության շրջանակներ՝ տեղական և գլոբալ:

**Question 5:** Գրեք Python ֆունկցիա sum\_digit, որը ընդունում է ոչ բացասական ամբողջ թիվ և վերադարձնում է նրա թվանշանների գումարը: (4 միավոր)

```
def sum_digit(n):  
    sum = 0  
    while n > 0:  
        digit = n % 10  
        sum += digit  
        n //= 10  
    return sum
```

**Question 6:** Գրեք Python ծրագիր, որը կարող է ընդունել 2-ից մեծ դրական ամբողջ թիվ և տպել, թե քանի անգամ պետք է բաժանել այս թիվը 2-ի, որպեսզի ստանանք 2-ից փոքր արժեք: (4 միավոր)

```
n = int(input("Enter a positive integer greater than 2: "))  
count = 0  
while n > 2:  
    n /= 2  
    count += 1  
print("Number of divisions:", count)
```

**Question 7:** Գրեք Python ֆունկցիա, որը ընդունում է 2D ցուցակ (մատրիցա) և վերադարձնում է նրա տրանսպոնացվածը: (8 միավոր)

```
def transpose(matrix):  
    rows = len(matrix)  
    cols = len(matrix[0])  
    transpose_matrix = [[0 for _ in range(rows)] for _ in range(cols)]  
    for i in range(rows):  
        for j in range(cols):
```

```
        transpose_matrix[j][i] = matrix[i][j]  
    return transpose_matrix
```

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

- <https://github.com/DeviLinux-sudo/Pre-Placement-Assignments>