

## 1. Ի՞նչ են թվաբանական օպերատորները Python-ում: (1 միավոր)

### Պատասխան:

Python-ում թվաբանական օպերատորները օգտագործվում են թվերի վրա տարբեր գործողություններ կատարելու համար: Այսպիսի օպերատորներ են՝

- + (գումարում)
- - (հանում)
- \* (բազմապատկում)
- / (բաժանում)
- // (ամբողջ բաժանում)
- % (մնացորդ)
- \*\* (ուժ)

## 2. Ի՞նչ է \$ֆունկցիան Python-ում և ինչո՞ւ են օգտագործվում \$ֆունկցիաները: (1 միավոր)

### Պատասխան:

Python-ում \$ֆունկցիան կոդի բլոկ է, որը կատարում է որոշակի գործառնություն: \$ֆունկցիաները օգտագործվում են՝

- **Կոդը մոդուլյարացնելու համար:** բարդ խնդիրները բաժանելու փոքր, վերօգտագործելի \$ֆունկցիաների:
- **Կոդի ընթերցելիությունը բարելավելու համար:** կոդը ավելի կազմակերպված և հասկանալի դարձնելու համար:
- **Կրկնվող կոդը նվազեցնելու համար:** նույն կոդը մի քանի անգամ գրելուց խուսափելու համար:
- **Կոդի վերօգտագործումը խթանելու համար:** \$ֆունկցիաները օգտագործելու ծրագրի տարբեր մասերում:

## 3. Ի՞նչ է + օպերատորի օգտագործման արդյունքը երկու ցուցակների վրա և + օպերատորի օգտագործման արդյունքը ցուցակի և ամբողջ թվի վրա: (1 միավոր)

### Պատասխան:

- **Երկու ցուցակների համար:** + օպերատորը կոնկատենացնում է երկու ցուցակները, ստեղծելով նոր ցուցակ, որը պարունակում է երկու ցուցակների բոլոր տարրերը:
- **Ցուցակի և ամբողջ թվի համար:** Այս գործողությունը Python-ում ուղղակիորեն չի աջակցվում: Դուք պետք է ամբողջ թիվը վերածեք մեկ տարր պարունակող ցուցակի և այնուհետև կոնկատենացնեք երկու ցուցակները:

## 4. Բացատրեք հավաքածուների (sets) հասկացությունը Python-ում: (1 միավոր)

### Պատասխան:

Հավաքածուները (sets) Python-ում ոչ կրկնվող տարրերի անկարգավոր հավաքածուներ են: Դրանք օգտագործվում են տարրերի գոյությունը արագ ստուգելու, կրկնությունները հեռացնելու և տարրերի միությունը, հատումը և տարբերությունը հաշվարկելու համար:

## 5. Օգտագործողից ստացեք տարրերի ցուցակը (բաժանված տարածությամբ) և տպեք նոր ցուցակ, որը ներմուծված ցուցակի հակադարձն է, առանց ներկառուցված \$ֆունկցիաներ օգտագործելու: (4 միավոր)

### Պատասխան:

```
def reverse_list(input_list):  
    reversed_list = []  
    for i in range(len(input_list) - 1, -1, -1):  
        reversed_list.append(input_list[i])  
    return reversed_list
```

```
input_str = input("Enter elements separated by spaces: ")
input_list = input_str.split()
reversed_list = reverse_list(input_list)
print(reversed_list)
```

**6. Գրեք Python ֆունկցիա, որը վերցնում է բառարանների ցուցակ և բանալի-արժեք զույգ, և վերադարձնում է նոր ցուցակ, որը պարունակում է միայն այն բառարանները, որոնք ունեն նշված բանալի-արժեք զույգը, օգտագործելով lambda ֆունկցիա: (4 միավոր)**

**Պատասխան:**

```
def filter_dictionaries(dictionaries, key, value):
    return list(filter(lambda d: d.get(key) == value, dictionaries))
```

**7. 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)**

**Պատասխան:**

```
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
```

- [https://github.com/abufaiz/30DaysOfPython\\_ArewaDataScience](https://github.com/abufaiz/30DaysOfPython_ArewaDataScience)
- <https://github.com/Piusin/Google-Africa-DEB-main>