

1. Tekrar Eden Sayıyı Bulma (List, Set)

- **Soru:** Bir liste içinde tekrar eden ilk sayıyı bulun.
- **İpucu:** `set` yapısını kullanarak daha hızlı çözüm bulunabilir.
- **Örnek:** `[2, 1, 2, 3]` girdisi için sonuç `2` olmalı.

2. Anagram Kontrolü (String, Dictionary)

- **Soru:** İki kelimenin anagram olup olmadığını kontrol eden bir fonksiyon yazın.
- **İpucu:** `Counter` veya `sorted()` kullanılabilir.
- **Örnek:** `("listen", "silent")` girdisi için sonuç `True` olmalı.

3. Dönüşümlü Liste Oluşturma (List)

- **Soru:** İki liste verildiğinde, bunları dönüşümlü olarak birleştiren bir fonksiyon yazın.
- **İpucu:** `zip()` fonksiyonu kullanılabilir.
- **Örnek:** `([1, 2, 3], [4, 5, 6])` girdisi için sonuç `[1, 4, 2, 5, 3, 6]` olmalı.

4. En Büyük Toplam Alt Dizi (List)

- **Soru:** Bir dizide ardışık elemanlardan oluşan en büyük toplamı bulan bir fonksiyon yazın.
- **İpucu:** Kadane algoritması kullanılabilir.
- **Örnek:** `[-2, 1, -3, 4, -1, 2, 1, -5, 4]` girdisi için sonuç `6` olmalı.

5. Sayıları Ayırma (List, Tuple)

- **Soru:** Bir listedeki tüm pozitif ve negatif sayıları ayırıp tuple içinde döndüren bir fonksiyon yazın.
- **İpucu:** Liste anlama (list comprehension) kullanılabilir.
- **Örnek:** `[1, -2, 3, -4]` girdisi için sonuç `([1, 3], [-2, -4])` olmalı.

6. Palindrome Kontrolü (String)

- **Soru:** Bir kelimenin palindrome olup olmadığını kontrol eden bir fonksiyon yazın.
- **İpucu:** String ters çevrilebilir.
- **Örnek:** `"madam"` girdisi için sonuç `True` olmalı.

7. Çok Boyutlu Listeyi Düzleştirme (List, Recursion)

- **Soru:** İç içe geçmiş çok boyutlu bir listeyi tek boyutlu bir listeye çeviren bir fonksiyon yazın.
- **İpucu:** Rekürsiyon kullanılabilir.
- **Örnek:** `[[1, 2], [3, [4, 5]]]` girdisi için sonuç `[1, 2, 3, 4, 5]` olmalı.

8. Tersine Çevirme (String)

- **Soru:** Bir stringi tersine çeviren bir fonksiyon yazın.
- **İpucu:** String dilimleme (slicing) kullanılabilir.
- **Örnek:** `"hello"` girdisi için sonuç `"olleh"` olmalı.

9. Eşsiz Sayıları Bulma (List, Set)

- **Soru:** Bir listede sadece bir kez görünen sayıları bulun.
- **İpucu:** `set` ve `list` yapıları kullanılabilir.
- **Örnek:** `[1, 2, 2, 3, 4, 4, 5]` girdisi için sonuç `[1, 3, 5]` olmalı.

10. Fibonacci Dizisi (Recursion, List)

- **Soru:** İlk `n` Fibonacci sayısını bulan bir fonksiyon yazın.
- **İpucu:** Rekürsiyon veya dinamik programlama ile çözülebilir.
- **Örnek:** `n = 5` için sonuç `[0, 1, 1, 2, 3]` olmalı.

11. En Sık Tekrar Eden Karakter (String, Dictionary)

- **Soru:** Bir stringde en sık tekrar eden karakteri bulun.
- **İpucu:** `dictionary` kullanılabilir.
- **Örnek:** `"aabbcc"` girdisi için sonuç `b` olmalı.

12. Sayıların Karesi (List, Function)

- **Soru:** Bir liste içindeki sayıların karesini hesaplayan bir fonksiyon yazın.
- **İpucu:** `map()` ve lambda fonksiyonu kullanılabilir.
- **Örnek:** `[1, 2, 3, 4]` girdisi için sonuç `[1, 4, 9, 16]` olmalı.

13. Birleştirilmiş Listeyi Sıralama (List, Sorting)

- **Soru:** İki listeyi birleştirip sıralayan bir fonksiyon yazın.
- **İpucu:** `sorted()` ve `extend()` kullanılabilir.
- **Örnek:** `([3, 1, 4], [2, 5])` girdisi için sonuç `[1, 2, 3, 4, 5]` olmalı.

14. Tarih Formatını Değiştirme (String)

- **Soru:** `YYYY-MM-DD` formatındaki bir tarihi `DD-MM-YYYY` formatına çeviren bir fonksiyon yazın.
- **İpucu:** `split()` ve `join()` kullanılabilir.
- **Örnek:** `"2023-12-31"` girdisi için sonuç `"31-12-2023"` olmalı.

15. Dizi Döndürme (List)

- **Soru:** Bir diziyi belirtilen adım kadar sağa döndüren bir fonksiyon yazın.
- **İpucu:** Dilimleme (slicing) kullanılabilir.
- **Örnek:** `([1, 2, 3, 4], 2)` girdisi için sonuç `[3, 4, 1, 2]` olmalı.