Genel Kurallar:

***Kod pratiğinizi Google Colab üzerinden yapabilirsiniz. Google arama butonunda yazdığınızda direkt karşınıza çıkacaktır. Gmail hesabınız giriş yaparak ile kendi Colab hesabınızı oluşturabilirsiniz.**

***Aşağıda belirttiğimiz açıklamalarda, # işareti ile başlayan kısımlar yazmanız gereken kodu açıklamaktadır.**

4.Gün: Listeler

1. Spor takımlarını analiz ediyorsunuz. Her bir takımın üyeleri bir listede bulunmaktadır. Koç, listede ilk isim olarak, kaptan ikinci isim olarak ve diğer oyuncular da onlardan sonra sıralı olarak bulunmaktadır. Bu listeler en başta en iyi takımlar olmak üzere sona doğru en kötü takıma doğru diğer başka bir listenin içerisinde bulunmaktadır. En kötü takımın kaptanını seçmek için aşağıdaki fonksiyonu tamamlayın.

```
# def losing_team_captain(teams):
    #buraya yukarıdaki bilgilere göre kendiniz bir isim list tanımlayıp, en kötü takım kaptanını döndüren algoritmayı yazın.
    pass
```

2. Mario Kart'ın son tekrarlaması Purple Shell adında ekstra sinir bozucu bir parça olacak. Kullanıldığında, sonuncu olarak yarışan yarışmacıyı ilk sıraya, ilk sırada yarışan yarışmacıyı da son sıraya taşımaktadır. Purple Shell etkisini uygulamak için aşağıdaki fonksiyonu tamamlayın.

```
Yarismaci_listesi = r = ["Mario", "Bowser", "Luigi"]

def purple_shell(racers):
    """ Yarışmacıların listesi verilmiştir, listenin başında bulunan ilk yarışmacıyı son sıraya yerleştirin ve sonra bunun tam tersini yapın. Aşağıda listin ilk hali ve fonksiyon sonrası olması gereken son hali örek olarak belirtilmiştir.
    >>> r = ["Mario", "Bowser", "Luigi"]
    >>> purple_shell(r)
    >>> r
    ["Luigi", "Bowser", "Mario"]
    """
    pass
```

3. Aşağıdaki listelerin uzunlukları nedir? Değişken uzunluklarını tahminlerinizle doldurun. (len() fonsiyonunu çağırmadan, her bir liste için tahminler yapmaya çalışın. Sonra len() ile yanıtlarınızı kontrol edin.)

a = [1, 2, 3]

b = [1, [2, 3]]

c = []

d = [1, 2, 3][1:]

4. Bir partiye katılan kişileri ve hangi sırayla katıldıklarını kaydetmek için listeler kullanıyoruz. Örneğin, sıradaki liste Adela'nın ilk sırada, Ford'un ise son sırada katıldığı bir partiyi temsil etmektedir.

party_attendees = ['Adela', 'Fleda', 'Owen', 'May', 'Mona', 'Gilbert', 'Ford']

Eğer bir davetli, partinin tüm davetlilerinin en az yarısından geç geldiyse, o kişi ''kibarca geç (fashionably late)'' sayılmaktadır. Ancak, en geç gelenlerden biri olmamak zorundadır (bu olayı daha ileriye taşımaktadır). Bu örneğin üzerine, Mona ve Gilbert ''kibarca geç (fashionably late)'' sayılan tek davetlilerdir.

Aşağıdaki, davetlilerin kişi bazında bulunduğu ve ''kibarca geç (fashionably late)'' olup olmadığını gösteren fonksiyonu tamamlayın.

```
def fashionably_late(arrivals, name):
    # Partiye gelenler isimleriyle birlikte düzenli bir sırada verilmiştir, gelen davetlinin ''kibarca geç (fashionably late)'' olup olmadığı sonucunu döndüren algoritmayı yazın.
    pass
```