

*BU SORULARI LABA GELMEDEN ÖNCE ÇÖZÜP, LABDA VAKİT KALIRSA  
ÇÖZÜMLERİNİZİ ASİSTANLARINIZLA DEĞERLENDİREBİLİRSİNİZ.*

---

Orta seviye dizi problemi için:

<https://www.hackerrank.com/challenges/array-left-rotation>

---

### **BİR DİZİDEKİ SAYILARIN KAÇ ADET TEKRARLANDIĞINI BULMAK.**

Kullanıcıdan bir dizi boyutu, aralığın min değeri ve max değeri okunup; verilen boyutta verilen aralıktaki rastgele sayıları içeren bir dizi oluşturulsun. Daha sonra da "Dizi oluşturuldu" mesajı basılıp kullanıcıdan bir değer alınsın. Bu değer verilen dizide var mı yok mu kontrol edilip varsa kaç kere olduğu ekrana yazılsın, yoksa "Bulunamadı" yazılsın.

Örnek:

- Dizi boyutu giriniz: 5
  - Aralık için min değer girin : 10
  - Aralık için max değer girin: 50
  - Dizi oluşturuldu !
  - Aranacak değer girin : 12
  - Dizide 3 tane 12 bulundu.
- 

### **ŞİFRELEME SİSTEMİ (Soru Absolute Java (by Walter Savitch) kitabından alıntı.)**

---

Traditional password entry schemes are susceptible to “shoulder surfing” in which an attacker watches an unsuspecting user enter his or her password or PIN number and uses it later to gain access to the account. One way to combat this problem is with a randomized challenge-response system. In these systems, the user enters different information every time based on a secret in response to a randomly generated challenge. Consider the following scheme in which the password consists of a five-digit PIN number (00000 to 99999). Each digit is assigned a random number that is 1, 2, or 3. The user enters the random numbers that correspond to their PIN instead of their actual PIN numbers.

For example, consider an actual PIN number of 12345. To authenticate it, the user would be presented with a screen such as the following:

```
PIN: 0 1 2 3 4 5 6 7 8 9
NUM: 3 2 3 1 1 3 2 2 1 3
```

The user would enter 23113 instead of 12345. This does not divulge the password even if an attacker intercepts the entry because 23113 could correspond to other PIN numbers, such as 69440 or 70439. The next time the user logs in, a different sequence of random numbers would be generated, such as the following:

```
PIN: 0 1 2 3 4 5 6 7 8 9
NUM: 1 1 2 3 1 2 2 3 3 3
```

Your program should simulate the authentication process. Store an actual PIN number in your program. The program should use an array to assign random numbers to the digits from 0 to 9. Output the random digits to the screen, input the response from the user, and output whether or not the user’s response correctly matches the PIN number.