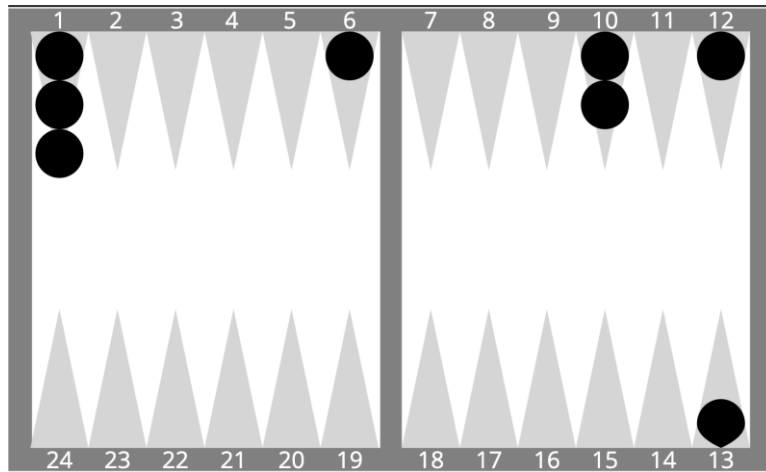


Problem:

Elimizde bir tavla tahtası var. Bu tahta üzerindeki her üçgen saat yönüne doğru numaralandırılmış ve üzerinde rastgele yerleştirilmiş pullarımız var. Hangi üçgende kaç tane pul olduğunu aşağıdaki gibi bir [sözlük](#) içerisinde anlayabiliyoruz:

```
checkers = {1: 3, 6: 1, 10: 2, 12: 1, 13: 1}
```

Bu sözlük aşağıdaki resimdeki gibi bir dağılımı gösteriyor:



Sizden istediğimiz pulların pozisyonlarını, ve atılan iki zarın değerini girdi olarak alan ve bulabildiği hamleleri bulup puanlayarak 0 dan büyük puanı olan hamleleri geri döndüren `find_moves` adında bir fonksiyon yazmanız.

Örnek Girdi

```
def find_moves(checkers, dice1, dice2)
    ...
    ...
    ...

print(find_moves(checkers, 6, 1))
```

Örnek Çıktı

Üçlü tüplerden oluşan bir liste çıkmasını bekliyoruz. İlk hamleyi gösteren bir tuple, ikinci hamleyi gösteren bir tuple, iki hamlenin yarattığı puanı gösteren bir integer.

```
>>> (((1, 7), (6, 7), 3), ..., ..., ...)
      +      +      +
      |      |      +--> Bu hamlenin kaç puan ettiği.
      |      +-----> 6 numaralı üçgendeki pulu 7 ye
      +-----> 1 numaralı üçgendeki pulu 7 ye
```

Hamlenin Puanlanması

- Önemli kapılar: 5, 6, 7, 8, 17, 18, 19, 20
- Bir kapı kapatmak +1 puan, eğer önemli kapı ise ise +2
- Bir kapı açmak -1 puan, eğer önemli kapı ise -2
- Bir taş açığa çıkarmak -1 puan.
- Bir taş açığından kurtarmak +1 puan.

Ek Not

Bu find_moves fonksiyonunu test eden [doctest](#) ya da [unittestler](#) yazabilerseniz ekstra puan sebebidir.