**Project Euler Problemleri**

1. We shall say that an n-digit number is pandigital if it makes use of all the digits 1 to n exactly once; for example, the 5-digit number, 15234, is 1 through 5 pandigital. The product 7254 is unusual, as the identity, 39 x 186 = 7254, containing multiplicand, multiplier, and product is 1 through 9 pandigital. Find the sum of all products whose multiplicand/multiplier/product identity can be written as a 1 through 9 pandigital. HINT: Some products can be obtained in more than one way so be sure to only include it once in your sum.

**Cevap:**

1. 1'den 9'a kadar tam dijital kimlik oluşturan tüm çarpan, çarpan ve çarpım kombinasyonlarını bulmamız gerekiyor.

2.Bunu yapmak için, çarpan ve çarpan için olası değerleri yineleyeceğiz.

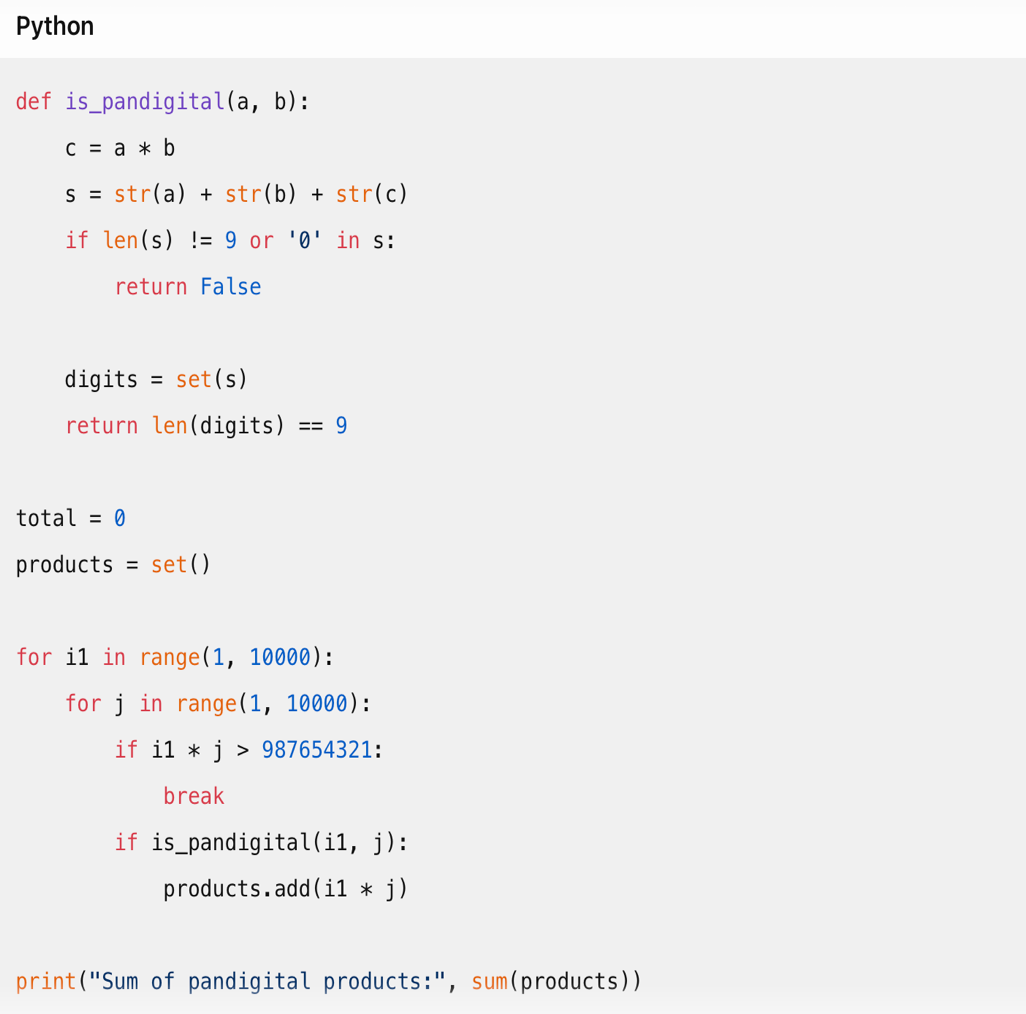
3. Her bir çift (i1, j) için çarpımı hesaplayacağız (toplam = i1 \* j).

4. i1, j ve sum'u tek bir dizede birleştireceğiz ve 1'den 9'a kadar tüm rakamları tam olarak bir kez içerip içermediğini kontrol edeceğiz.

5.Eğer öyleyse, toplamı toplamımıza ekleyeceğiz.

6.Son olarak her ürünü toplamımıza yalnızca bir kez dahil ettiğimizden emin olacağız.

İşte bu sorunu çözecek bir Python uygulaması:



1. The following iterative sequence is defined for the set of positive integers:

nn/2 (n is even)

n3n+1 (n is odd)

Using the rule above and starting with 13, we generate the following sequence:

1340201051684→2→1.

It can be seen that this sequence (starting at 13 and finishing at 1) contains 10 terms. Although it has not been proved yet (Collatz Problem), it is thought that all starting numbers finish at 1.

Which starting number, under one million, produces the longest chain?

NOTE: Once the chain starts the terms are allowed to go above one million.

**Cevap:**

Collatz problemini Python kullanarak çözelim. 3x + 1 problemi veya dolu dizisi olarak da bilinen Collatz varsayımı bir başlangıç ​​numarası alır. Çift ise 2'ye bölünür, tek ise 3 ile çarpılır ve 1 eklenir. Bu işlem 1'e ulaşana kadar tekrarlanır. Her ne kadar tüm sayılar için kanıtlanmamış olsa da, tüm başlangıç ​​sayılarının eninde sonunda ulaştığına inanılmaktadır.

İşte bu sorunu çözecek bir Python uygulaması:

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. The sum of the primes below 10 is 2+3+5+ 7 = 17.

Find the sum of all the primes below two million.

**Cevap**:

Asal sayıları üretmenin etkili bir yolu olan Eratosthenes Eleği algoritmasını kullanacağız.

İşte bu sorunu çözecek bir Python uygulaması:

metin, ekran görüntüsü, yazı tipi, cebir içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be: 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.

**Cevap:**

1. İlk olarak, bir Fibonacci dizisi oluşturacağız. Bu dizideki terimler, önceki iki terimin toplamıyla elde edilir.
2. Başlangıçta a = 0 ve b = 1 olarak ayarlayacağız.
3. Döngü içinde, b değeri n (dört milyon) değerini aşmadığı sürece devam edeceğiz.
4. Her adımda, a ve b değerlerini güncelleyeceğiz:
   * a değeri b olacak.
   * b değeri a + b olacak.
5. Eğer b çift sayı ise, bu terimi toplamımıza ekleyeceğiz.
6. Döngü sona erdiğinde, toplamı döndüreceğiz.

Python kodu şu şekildedir:

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. If p is the perimeter of a right angle triangle with integral length sides, (a, b, c), there are exactly three solutions for p = 120. {20, 48, 52), (24, 45, 51), (30, 40, 50}

For which valde of p≤ 1000, is the number of solutions maximised?

**Cevap:**

Tam kenar uzunluklarına sahip bir dik açılı üçgenin çözüm sayısını maksimuma çıkaracak şekilde (p) (çevre) değerini arıyoruz.

1. Öncelikle, dik açılı bir üçgen için Pisagor teoreminin geçerli olduğunu biliyoruz: (a^2 + b^2 = c^2).

2. Çevre (p = a + b + c) ile verilir.

3. p \leq 1000) için maksimum çözüm sayısını bulmak istiyoruz.

4. p)'nin olası değerleri üzerinden yineleyeceğiz ve kaç geçerli çözümün mevcut olduğunu kontrol edeceğiz.

Cevabı bulmak için Python kodu:

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. The number, 197, is called a circular prime because all rotations of the digits: 197, 971, and 719, are themselves prime. There are thirteen such primes below 100: 2, 3, 5, 7, 11, 13, 17, 31, 37, 71, 73, 79, and 97. How many circular primes are there below one million?

**Cevap:**

1. İlk olarak, is\_prime(n) fonksiyonu verilen bir sayının asal olup olmadığını kontrol eder. Eğer sayı 2’den küçükse veya böleni varsa (2’den büyük ve karekökünden küçük olan sayılara böleni bakılır), asal değil olarak döner. Aksi takdirde asal olarak kabul edilir.
2. rotate(n) fonksiyonu, bir sayının rakamlarını döndürür. Örneğin, 197 sayısının döndürülmüş hali 971 olur.
3. is\_circular\_prime(n) fonksiyonu, verilen bir sayının dairesel asal olup olmadığını kontrol eder. İlk olarak, sayının asal olup olmadığına bakar. Eğer asal değilse, dairesel asal değildir. Eğer asalsa, sayının döndürülmüş hali üzerinden döngüye girer ve tüm döndürülmüş hallerinin asal olup olmadığını kontrol eder. Eğer bir döndürülmüş hali asal değilse, dairesel asal değildir.
4. Son olarak, circular\_primes listesi, 2 ile 1 milyon arasındaki sayılar üzerinde is\_circular\_primefonksiyonunu kullanarak dairesel asal sayıları bulur.

Python kodu şu şekildedir:

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. The prime 41, can be written as the sum of six consecutive primes:

412+3+5+7+11+13.

This is the longest sum of consecutive primes that adds to a prime below one-hundred.

The longest sum of consecutive primes below one-thousand that adds to a prime, contains 21 terms, and is equal to 953.

Which prime, below one-million, can be written as the sum of the most consecutive primes?

**Cevap:**

1.Bir asal sayı ile iki katın karesinin toplamı olarak ifade edilemeyen tek bileşik sayıyı (1'den büyük) arıyoruz.

2.Verdiğiniz örneklerin (9, 15, 21, 25, 27 ve 33) hepsi şu modeli takip ediyor: tek bileşik = asal + 2 \* kare.

3.Peki ya 1 numara? Kompozit olmadığı için kalıba uymuyor. Aslında 1 bir birimdir ve ne asal ne de bileşiktir.

4.Şimdi Goldbach'ın diğer varsayımını sağlamayan en küçük tek bileşik sayıyı bulalım.

Python kodu şu şekildedir:

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. f we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23. Find the sum of all the multiples of 3 or 5 below 1000.

**Cevap:**

1000’in altındaki 3 veya 5’in katlarının toplamını hesaplayan Python kodu:

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

1. Using names.txt (right click and 'Save Link/Target As...'), a 46K text file containing over five-thousand first names, begin by sorting it into alphabetical order. Then working out the alphabetical value for each name, multiply this value by its alphabetical position in the list to obtain a name score. For example, when the list is sorted into alphabetical order, COLIN, which is worth 3+15+12+9+14=53, is the 938th name in the list. So, COLIN would obtain a score of 938 x 53 = 49714. What is the total of all the name scores in the file?

**Cevap:**

1.İsimleri dosyadan okuyun: İlk olarak, name.txt dosyasındaki isimleri okumamız gerekiyor. Dosyayı verilen bağlantıdan indirebilirsiniz.

2.İsimleri alfabetik olarak sıralayın: İsimleri aldıktan sonra bunları alfabetik sıraya göre sıralayacağız.

3.İsim puanlarını hesaplayın: Her ismin alfabetik değerini (harf konumlarının toplamı) hesaplayacağız ve ardından bunu sıralı listedeki konumuyla çarpacağız.

İ4.sim puanlarını toplayın: Son olarak toplamı elde etmek için tüm isim puanlarını toplayacağız**.**

Python kodu şu şekildedir:

**metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

1. The prime factors of 13195 are 5, 7, 13 and 29.What is the largest prime factor of the number 600851475143?

**Cevap:**

* largest\_prime\_factor fonksiyonu, verilen sayının en büyük asal bölenini bulmak için etkili bir yöntem kullanıyor. Bu fonksiyon, faktörleri 2’den başlayarak kontrol eder ve bölenleri bulduğunda sayıyı böler. Bu sayede en büyük asal böleni elde ederiz.
* factor \* factor <= n koşulu, faktörün karesinin verilen sayıdan küçük veya eşit olması gerektiğini belirtir. Bu, faktörleri kontrol etmek için gereken minimum aralığı daraltır ve performansı artırır.
* n //= factor ifadesi, sayının faktöre bölünmesini gerçekleştirir. Böylece aynı faktörü birden fazla kez kullanabiliriz.

Python kodu şu şekildedir:

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Ebrar Cengiz

2203073