

---

# Programación para la Computación Científica - IA



## Python for Programmers

**Numpy II**

Universidad Sergio Arboleda  
*Prof. John Corredor*

---

## Encontrando Factores Primos

Los factores primos son números primos que dividen un número entero exactamente sin resto. Encontrar factores primos parece casi imposible de descifrar. Sin embargo, utilizando el algoritmo correcto, el método de factorización de Fermat y NumPy: se vuelve muy fácil. La idea es factorizar un número  $N$  en dos números  $c$  y  $d$ , de acuerdo con la siguiente ecuación:

$$N = cd = (a + b)(a - b) = a^2 - b^2$$

## Encontrando Factores Primos

El algoritmo requiere que intentemos varios valores de prueba para  $a$ .

1.- Crear una matriz de valores de prueba

```
a = numpy.ceil(numpy.sqrt(n))
```

```
lim = min(n, LIM)
```

```
a = numpy.arange(a, a + lim)
```

```
b2 = a ** 2 - n
```

(Usamos la función `ceil` para devolver el techo de la entrada, en cuanto a elementos.)

2.- Obtenga la parte fraccionaria de la matriz b.

```
fractions = numpy.modf(numpy.sqrt(b2))[0]
```

3.- Encuentra 0 fracciones.

```
indices = numpy.where(fractions == 0)
```

4.- Encuentra la primera aparición de una fracción cero.

```
a = numpy.ravel(numpy.take(a, indices))[0]
```

(En realidad, solo necesitamos la primera aparición de una fracción cero. Primero, llame a la función de toma NumPy con la matriz de índices del paso anterior para obtener los valores de fracciones cero. Ahora necesitamos "aplanar" esta matriz con la función NumPy **ravel**)

```

import numpy as np
#The prime factors of 13195 are 5, 7, 13 and 29.
#What is the largest prime factor of the number 600851475143 ?
N = 600851475143
LIM = 10 ** 6
def factor(n):
    #1. Create array of trial values
    a = np.ceil(np.sqrt(n))
    lim = min(n, LIM)
    a = np.arange(a, a + lim)
    b2 = a ** 2 - n
    #2. Check whether b is a square
    fraction, b = np.modf(b2)
    #3. Find indices
    indices = np.where(b == 0)
    #4. Find
    a = np.r_[a[indices], 1]
    a = int(a)
    b = np.sort(a)
    b = int(b)
    c = a + b
    d = a - b

    if c == 1 or d == 1:
        return
    print(c, d)
    factor(c)
    factor(d)

factor(N)

```

**Function****Description**

ceil

Calculates the ceiling of array elements.

modf

Returns the fractional and integral part of floating point numbers.

where

Returns array indices based on condition.

ravel

Returns a flattened array.

take

Takes element from an array.

## Encontrando Numeros Palindricos

Un número palindrómico se lee igual en ambos sentidos. El palíndromo más grande hecho del producto de dos números de 2 dígitos es  $9009 = 91 \times 99$ . Tratemos de encontrar el palíndromo más grande hecho del producto de dos números de 3 dígitos.

1. Cree una matriz de números de 3 dígitos.

```
a = numpy.arange(100, 1000)
numpy.testing.assert_equal(100, a[0])
numpy.testing.assert_equal(999, a[-1])
```

(Verifique el primer y último elemento de la matriz con la función `assert_equal` del paquete `numpy.testing`)

### 2. Cree el array de productos

```
numbers = numpy.outer(a, a)
numbers = numpy.ravel(numbers)
numbers.sort()
numpy.testing.assert_equal(810000, len(numbers))
numpy.testing.assert_equal(10000, numbers[0])
numpy.testing.assert_equal(998001, numbers[-1])
```

(se crea una matriz para contener todos los productos posibles de los elementos de la matriz de 3 dígitos consigo mismo. Podemos lograr esto con la función externa. La matriz resultante debe aplanarse con **ravel**, para poder iterar fácilmente sobre ella. Llame al método **sort** en la matriz para asegurarse de que la matriz esté ordenada correctamente.)

## Challenge 01

1. Encuentra el palíndromo más grande hecho del producto de dos números de 3 dígitos.





# References

- ★ Kernighan, Brian W., and Dennis M. Ritchie. The C Programming Language. Vol. 2. Englewood Cliffs: prentice-Hall, 1988.
- ★ Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. Operating System Concepts. Vol. 8. Wiley, 2013.
- ★ <https://planningtank.com/computer-applications/data-processing-cycle>
- ★ <https://www.talend.com/resources/what-is-data-processing/>
- ★ <https://peda.net/kenya/ass/subjects2/computer-studies/form-3/data-processing/dpc2>
- ★ [https://www.academia.edu/38210518/What\\_is\\_Data\\_Processing](https://www.academia.edu/38210518/What_is_Data_Processing)
- ★ <https://www.studocu.com/en/document/polytechnic-university-of-the-philippines/information-and-communication-technology/lecture-notes/data-processing-lectures-in-data-processing/3167716/view>
- ★ <http://download.nos.org/srsec330/330L2.pdf>