



## Requerimientos

- Python 3

## Ejecución

1. Clonar el repositorio

```
$ git clone git@github.com:Adr550/Lab_Analisis_C.git  
$ cd Lab_Analisis_C
```

1. Ejecutar el archivo main.py

```
python main.py
```

## EJERCICIO 1

**Instrucciones:** Realice un programa en Python, donde genere las sucesiones decimales de 4 dígitos, donde el 0 está permitido colocarlo de lado izquierdo.

a) Genere las sucesiones de 4 dígitos diferentes e imprima su conteo.

```
#1) Contar Sucesiones donde se vale repetir  
def sucesionesNormal(N,r):  
    result=len(A)  
    return result  
  
#2) Sucesion de 4 dígitos diferentes  
def sucesionesDiferentes(N,r):  
    A=list(iter.permutations(N,r))  
    result=len(A)  
    return result
```

### Respuesta

Tiene como resultado: 10000 posibles combinaciones.  
El Resultado de 4 dígitos distintos en una sucesión es: 5040.

b) Genere las sucesiones de 4 dígitos que contengan uno o más dígitos repetidos e imprima su conteo.

```
def sucesionRepetidos(N,r):  
    total=sucesionesNormal(N,r) - sucesionesDiferentes(N,r)  
    return total
```

### Respuesta

Una sucesión con al menos un repetido tiene: 4960 posibilidades.

c) Imprima el conteo de elementos por cada una de las siguientes divisiones, utilizando la generación que realizó en el inciso b):

1. Se repite el dígito 4 veces.

```
def sucesionesTodosRepetidos(A):  
    contador = 0  
    for s in A:  
        # Comparamos si todos los elementos son iguales al primero  
        if s[0] == s[1] == s[2] == s[3]:  
            contador += 1  
    return contador
```

### Respuesta

Una sucesión con todos los dígitos repetidos es de: 10 posibilidades

2. Se repiten dos dígitos dos veces cada uno.

```
def sucesionesDoblePareja(lista):  
    contador = 0  
    for s in lista:  
        # 1. Convertimos la tupla a set para ver cuántos dígitos distintos hay
```

```

    distintos = set(s)
    # 2. Si hay exactamente 2 dígitos distintos...
    if len(distintos) == 2:
        # 3. Verificamos que sea "Doble Pareja" (2 y 2) y no "Trío" (3 y 1)
        # Tomamos cualquiera de los dos dígitos y contamos cuántas veces
aparece
        primer_digito = list(distintos)[0]

        if s.count(primer_digito) == 2:

            contador += 1
    return contador

```

### Respuesta

Una sucesión con 2 parejas de números tiene: 270 resultados.

3. Se repite un elemento dos veces y los otros no se repiten.

```

#4.3)una única pareja
def succionUnicaPareja(lista):
    contador = 0
    for s in lista:
        # 1. Convertimos la tupla a set para ver cuántos dígitos distintos hay
        distintos = set(s)

        if len(distintos) == 3:
            contador += 1
    return contador

```

### Respuesta

Una sucesión con una única pareja tiene: 4320 resultados.

4. Se repite un dígito 3 veces y el otro no se repite.

```

# se repite un dígito tres veces
def succionTresRepetidos(lista):
    contador = 0
    for s in lista:
        # 1. Convertimos la tupla a set para ver cuántos dígitos distintos hay

```

```

distintos = set(s)
if len(distintos) == 2:
    numero_a_probar = list(distintos)[0]
    if s.count(numero_a_probar) == 3 or s.count(numero_a_probar) == 1:
        contador += 1
return contador

```

### Respuesta

Una sucesión con tres dígitos repetidos es: 360.

## EJERCICIO 2

**Instrucciones:** Realizar el conteo de todos los números primos menores que 100, usando el principio de inclusión-exclusión y generando la lista o conjuntos de los elementos correspondientes para el cálculo y luego muestre el listado de todos los números primos menores que 100, usando algunos listas o conjuntos que usó en el cálculo anterior.

```

numeros_primos = [2]
todo = 100
numeros = [i for i in range(1,101,2)]

C1 = [i for i in numeros if i % 7 == 0 and i != 7 ]
C2 = [i for i in numeros if i % 3 == 0 and i != 3]
C3 = [i for i in numeros if i % 5 == 0 and i != 5]

def interseccion (multiplos1,multiplos2):
    """Esta función reúne los múltiplos en común"""
    interseccion = []
    for i in multiplos1 and multiplos2:
        if i in multiplos1 and multiplos2:
            interseccion.append(i)
    return interseccion

N1 = interseccion(C1,C2)
N2 = interseccion(C2,C3)
N3 = interseccion(C1,C3)

#Encontrar los numeros primos
for i in numeros:
    if i not in C1 and i not in C2 and i not in C3 and i != 1:
        numeros_primos.append(i)

```

```
else:
    continue
```

### Respuesta

La sucesión de números primos menores a 100, consta de 25 elementos

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67,
71, 73, 79, 83, 89, 97]
```

## EJERCICIO 3

**Instrucciones:** ¿Cuál es el valor de k después de ejecutar el siguiente código?

```
n = 7
m = 5
k = 0

for i1 in range(1, n+1):
    for i2 in range(1, i1+1):
        for i3 in range(1, i2+1):
            for i4 in range(1, i3+1):
                for i5 in range(1, i4+1):
                    k = k + 1

print(k)
```

Forme un conjunto cuyos elementos son las tuplas de los valores de las variables i5, i4, i3, i2 y i1 (en ese orden). Además, genere un conjunto de las muestras de 5 elementos de 7 elementos con reemplazo. Compare los dos elementos. ¿Qué concluye? (colocarlo como un comentario).

```
# Generar muestras de 5 elementos de 7 elementos con reemplazo

muestras_con_reemplazo = set(iter.combinations_with_replacement(range(1, 8),
5))
# este es el conjunto de las muestras de 5 elementos de 7 elementos con
reemplazo
```

### Respuesta

El valor k obtenido es de 462

Número de tuplas en el conjunto: 462

Número de muestras con remplazo: 462

Ya que ambos valores son iguales, se concluye que, el código genera exactamente las combinaciones con repetición de 5 elementos tomados de (1, 2, 3, 4, 5, 6, 7), que son 462 en total.

## Capturas de la ejecución

```
$ python main.py
```

```
=====
PARTE 1
=====
```

a. Contar una sucesión de 4 dígitos

Tiene como resultado: 10000 posibles combinaciones.

El Resultado de 4 dígitos distintos en una sucesión es: 5040.

b. Sucesiones con uno o más dígitos repetidos

Una sucesión con al menos un repetido tiene: 4960 posibilidades.

c. Conteo de elementos por cada una de las siguientes divisiones,  
utilizando la generación que realizó en el inciso b):

c.1. Todos repetidos

Una sucesión con todos los dígitos repetidos es de: 10 posibilidades.

c.2. Se repiten 2 dígitos 2 veces cada uno

Una sucesión con 2 parejas de números tiene: 270 resultados.

c.3. una única pareja

Una sucesión con una única pareja tiene: 4320 resultados.

c.4. se repite un dígito tres veces

Una sucesión con tres dígitos repetidos es: 360.

```
=====
PARTE 2
=====
```

Encontrar numeros primos usando el principio de inclusion y exclusion

Sucesión de primos: [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43,  
47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

Coneteo: 25

```
=====
PARTE 3
=====
```

Valor de k: 462

Número de tuplas en el conjunto: 462

Número de muestras con remplazo: 462

¿Son iguales los conjuntos? True

Las combinaciones con repetición de 5 elementos tomados de (1, 2, 3, 4, 5  
, 6, 7), que son 462 en total.

