

CONCEPTOS

- Pasaje BINARIO A DECIMAL X
- Pasaje DECIMAL A BINARIO V
- NÚMEROS PRIMOS } DIVISORES
- NÚMEROS PERFECTOS }
- FACTORIAL
- FIBONACCI Y NÚM DE ORO
- RAÍZ CUADRADA EXACTA
- DAR VUELTA UN NÚMERO
- SERIES Y POSICIONES
- MÚLTIPLO DE OTTO NÚMERO \rightarrow = divisible
- Mayor y menor
- Impar
- ejercicios contadores y acumuladores
- Ingreso hasta parar
- Números random
- Promedio y Porcentaje
- Números enteros
- Sumas sucesivas \rightarrow Producto $a * b \rightarrow 4 * 3 = \underbrace{4 * 4 * 4}_{3 \text{ veces}}$
- Números amigos
- Números deficientes \rightarrow suma de sus divisores $<$ al N^o
- si mismo

CONCEPTOS

múltiplo $\rightarrow \text{num} \% 3 == 0$ (ver: num es múltiplo de 3)

mayor y menor \rightarrow empieza $\begin{cases} \text{mayor} = \text{num} \\ \text{menor} = \text{num} \end{cases}$

se queda con
el primer ingreso
si es el mismo valor

```
if num > mayor:
    mayor = num
    ...
```

dar vuelta numero \rightarrow

```
num = 53781
```

```
for i in range(1, 4):
```

```
a = (a % 10000) * 10 37810
```

```
b = a // 10000 5
```

```
a = a + b 37810 + 5
```

```
Print(a) 37815 (1)
```

```
78153 (2)
```

```
81537 (3)
```

```
15378 (4)
```

} veces

factorial \rightarrow

```
num = ...
```

```
factorial = 1
```

```
while num >= 1:
```

```
    factorial = factorial * num
```

```
    num -= 1
```

```
Print(factorial)
```

número en un rango
o sino pedir otro

```
[ num = int(input())
  while num < 2 or num > 10:
    num = int(input())
```

Impar $\rightarrow \text{num} \% 2 != 0$ & $\text{num} // 2 == 1$

fibonacci →

```
a = 0
b = 1
cont = 1
```

(suma el resultado
y el num anterior)

mejor con
for *

```
while cont < 20:
    c = a + b
    print(c)
    a = b
    b = c
    cont += 1
```

→ posición

num de oro es dividir
resultado (b) y el
anterior (a)

↓
oro = b / a

Primos →

cantidad de divisores = 2 (divisible por 1 y a mismo)
si tiene 2 divisores es primo

```
cont = 0
divisor = 1
```

```
while divisor <= num:
    if num % divisor == 0:
        cont += 1
    divisor += 1
```

```
if cont > 2:
    print("no primo")
else:
    print("primo")
```

* con for empezar → for i in range(2, 21)

con while el contador = 1

while contador <= 25

0 1 1 2 3 5

0 1 2 3 4 5 → posición

Raíz cuadrada exacta

HOJA N°

FECHA

Resto empieza en 1 y se le van sumando +2

$$\begin{array}{r} 9 - 1 = 8 \\ \quad +2 \\ 8 - 3 = 5 \\ \quad +2 \\ 5 - 5 = 0 \end{array} \left. \vphantom{\begin{array}{r} 9 \\ 8 \\ 5 \end{array}} \right\} \text{CONT} = 3 \rightarrow \sqrt{9} = 3$$

0 \rightarrow raíz cuadrada exacta

$$4 - 1 = 3$$

$$\quad +2 \\ 3 - 3 = 0$$

$$\quad +2 \\ 0 - 5 = -5$$

-5 \rightarrow raíz no exacta
(valor < cero)

variables

- resto = 1 (empieza siempre en 1)
- cont = 0 (contador del resultado de la raíz)
- i = 0

Proceso

```

while i < 50:
    valor = random.randint(.....,.....)
    aux = valor

    while valor > 0:
        valor = valor - resto
        resto += 2
        cont += 1

    if valor == 0:
        print("tiene raíz cuadrada exacta")
        print("la raíz de", aux, "es:", cont)

    i += 1
    
```

Si valor termina siendo menor a cero no es exacta