

Soluciones a los problemas: Competición 1 Problemas matemáticos sencillos

Competición 1 Problemas matemáticos sencillos

¿Termina en 3?

```
# Leer un valor desde la entrada
n = input()

# Verificar si el ultimo caracter es "3"
if n[-1] == "3":
    print("SI") # Imprimir SI si termina en 3
else:
    print("NO") # Imprimir NO en caso contrario
```

Forma de diamante

```
# Leer un numero desde la entrada
n = int(input())

# Calcular la mitad
mitad = n // 2
espacios = mitad

# Parte superior del rombo
for i in range(mitad + 1):
    print(" " * espacios + "*" * (2 * i + 1)) # Imprimir espacios y asteriscos
    espacios -= 1

# Parte inferior del rombo
espacios = 1
for i in range(mitad - 1, -1, -1):
    print(" " * espacios + "*" * (2 * i + 1)) # Imprimir espacios y asteriscos
    espacios += 1
```

Suma de números naturales

```
# Leer un numero desde la entrada
n = int(input())

# Calcular e imprimir la suma de los primeros n numeros
print(n * (n + 1) // 2)
```

Sumitomo Century

```
# Leer un numero, calcular su centena y imprimirla
print((int(input()) - 1) // 100 + 1)
```

Comprobar si un número es primo (Optimizado)

```
# Leer un numero desde la entrada
n = int(input())

# Verificar si el numero es primo
for i in range(2, int(n**0.5) + 1):
    if n % i == 0:
        print("No es primo") # Imprimir si no es primo
        break
else:
    print("Es primo") # Imprimir si es primo
```

Suma de divisores optimizada

```
# Funcion para calcular la suma de los divisores de un numero
def suma_divisores(num):
    if num == 1:
        return 0 # 1 no tiene divisores propios
    suma = 1 # 1 es divisor de todos los numeros > 1
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            suma += i
            if num // i != i:
                suma += num // i # divisor complementario
    return suma

# Leer un numero desde la entrada
n = int(input())

# Imprimir la suma de divisores de cada numero desde 1 hasta n
for i in range(1, n + 1):
    print(f"{i}: {suma_divisores(i)}")
```

Factorización en primos

```
# Criba de Eratostenes para encontrar numeros primos hasta N
N = 1000001
dp = [True] * (N + 1)
dp[0] = dp[1] = False
```

```
for i in range(2, int(N**0.5) + 1):
    for j in range(i * i, N, i):
        dp[j] = False

# Guardar primos en una lista
p = []
for i in range(2, N):
    if dp[i]:
        p.append(i)

# Factorizacion prima del numero ingresado
pos = 0
n = int(input())
while pos < len(p) and n >= p[pos]:
    if n % p[pos] == 0:
        con = 0
        while n % p[pos] == 0:
            n //= p[pos]
            con += 1
        print(f"{p[pos]}^{con}") # Imprimir factor primo y su exponente
        pos += 1

# Si queda un factor primo mayor que la raiz
if n > 1:
    print(f"{n}^1")
```

Cajero del Mundo Binario

```
# Leer un numero desde la entrada
n = int(input())

sol = 0 # Contador de bits en 1

# Revisar cada bit hasta el bit 31
for i in range(32):
    if n & (2**i): # Comprobar si el bit i esta en 1
        sol += 1

print(sol) # Imprimir la cantidad de bits en 1
```

Reverso del número

```
# Leer un numero desde la entrada como cadena
n = input()

# Invertir la cadena
n = n[::-1]

# Convertir de nuevo a entero
```

```
n = int(n)

# Imprimir el numero invertido
print(n)
```

Sincronización de eventos — MCM aplicado

```
# Leer dos numeros desde la entrada
a, b = list(map(int, input().split()))

ab = a * b # Guardar el producto para calcular mcm luego

# Algoritmo de Euclides para calcular el mcd
while a != 0:
    a, b = b % a, a

# Calcular e imprimir el mcm usando el mcd
print(ab // b)
```

Números abundantes hasta N

```
# Funcion para determinar si un numero es abundante
def es_abundante(x):
    suma = 0
    for i in range(1, x):
        if x % i == 0:
            suma += i # Sumar divisores propios
    return suma > x # True si la suma de divisores es mayor que el numero

# Leer un numero desde la entrada
n = int(input())

solucion = []

# Buscar todos los numeros abundantes hasta n
for i in range(1, n + 1):
    if es_abundante(i):
        solucion.append(i)

# Imprimir resultados
if len(solucion) == 0:
    print("Ninguno")
else:
    print(", ".join(map(str, solucion)))
```