

Soluciones a los problemas: Clase 3 – Introducción y Fundamentos



Problemas: Clase 3 – Máximo común divisor (MCD) y mínimo común múltiplo (MCM)

¿Es divisible?

```
# Leer un número
n = int(input())
# Leer otro número
m = int(input())
# Muestra True si n es divisible por m, sino muestra False
print(n % m == 0)
```

Contar divisores

```
# Pide un número
n = int(input())
# Contador de divisores
sol = 0
# Límite hasta la raíz cuadrada
limite = int(n**(1/2)) + 1
for i in range(1, limite):
    # Si i divide a n
    if n % i == 0:
        sol += 1      # cuenta i
        if n // i != i: # si el otro divisor es distinto de i
            sol += 1  # cuenta también el otro divisor
# Muestra cuántos divisores tiene n
print(sol)
```

Máximo Común Divisor (MCD) — Fuerza Bruta

```
# Pide dos números
n, m = list(map(int, input().split()))
# Ciclo desde el mínimo hasta 1
for i in range(min(n, m), 0, -1):
    # Si i divide a los dos
    if n % i == 0 and m % i == 0:
        print(i) # muestra i
        break   # termina el ciclo
```

Máximo Común Divisor (MCD) — Algoritmo de Euclides

```
# Pide dos números
a, b = list(map(int, input().split()))
# Mientras a no sea 0
while a != 0:
    # Actualiza a y b
    a, b = b % a, a
# Imprime el MCD
print(b)
```

Mínimo Común Múltiplo (MCM) usando MCD

```
# Pide dos números
a, b = list(map(int, input().split()))
# Guarda el producto de a y b
ab = a * b
# Mientras a no sea 0
while a != 0:
    # Actualiza a y b
    a, b = b % a, a
# Muestra el mínimo común múltiplo
print(ab // b)
```

Simplificar fracción

```
# Pide dos números
a, b = list(map(int, input().split()))
# Guarda los valores originales
a1, b1 = a, b
# Mientras a no sea 0
while a != 0:
    # Actualiza a y b
    a, b = b % a, a
# Muestra los números divididos por su MCD
print(a1 // b, b1 // b)
```

Sincronización de eventos — MCM aplicado

```
# Pide dos números
a, b = list(map(int, input().split()))
# Guarda el producto de a y b
ab = a * b
# Mientras a no sea 0
while a != 0:
```

```
# Calcula el MCD
a, b = b % a, a
# Muestra el MCM (mínimo común múltiplo)
print(ab // b)
```

MCD de un arreglo de números

```
# Función para calcular el máximo común divisor (MCD) de dos números
def calcular_mcd(a, b):
    while a != 0:          # Mientras a no sea 0
        a, b = b % a, a    # Actualiza a y b
    return b               # Devuelve el MCD
# Pide la cantidad de números
n = int(input())
# Pide la lista de números
l = list(map(int, input().split()))
# Inicializa el MCD con el primer número
mcd = l[0]
# Calcula el MCD de toda la lista
for e in l[1:]:
    mcd = calcular_mcd(mcd, e)
# Muestra el MCD de todos los números
print(mcd)
```