

# Soluciones a los problemas: Clase 4 – Números Perfectos, Abundantes y Deficientes

---



## Problemas: Clase 4 – M Números Perfectos, Abundantes y Deficientes

### Números Perfectos Muy Sencillo

```
# Leer número
n = int(input())

# Comprobar si es 6 es perfecto
if n == 6:
    print("Perfecto")
else: # Si es 5 no es perfecto
    print("No es perfecto")
```

### Determinar si un número es Perfecto

```
# Leer número
n = int(input())

# Sumar divisores propios
su = 0
for i in range(1, n):
    if n % i == 0:
        su += i

# Comprobar si es perfecto
if su == n:
    print("Perfecto")
else:
    print("No es perfecto")
```

### Perfecto, Abundante o Deficiente

```
# Leer número
n = int(input())

# Sumar divisores propios
su = 0
for i in range(1, n):
    if n % i == 0:
        su += i
```

```
# Si la suma es igual a n → perfecto
if su == n:
    print("Perfecto")
# Si la suma es menor que n → deficiente
elif su < n:
    print("Deficiente")
# Si la suma es mayor que n → abundante
else:
    print("Abundante")
```

## Lista de números: Perfecto, Abundante o Deficiente

```
# Función que clasifica un número
def es_perfecto_abundante_deficiente(x):
    # Sumar divisores propios
    su = 0
    for i in range(1, x):
        if x % i == 0:
            su += i
    # Comparar la suma con el número
    if su == x:
        return "Perfecto"      # Igual → perfecto
    elif su < x:
        return "Deficiente"    # Menor → deficiente
    else:
        return "Abundante"     # Mayor → abundante

# Leer cantidad de números
n = int(input())

# Leer lista de números
li = list(map(int, input().split()))

# Imprimir resultado para cada número
for x in li:
    print(es_perfecto_abundante_deficiente(x))
```

## Números abundantes hasta N

```
# Función que comprueba si un número es abundante
def es_abundante(x):
    suma = 0
    for i in range(1, x):
        if x % i == 0:
            suma += i
    if suma > x:
        return True
    return False
```

```
# Leer número límite
n = int(input())

# Lista para guardar los números abundantes
solucion = []

# Comprobar del 1 al n
for i in range(1, n + 1):
    if es_abundante(i):
        solucion.append(i)

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

## Suma de divisores optimizada

```
# Calcula la suma de los divisores propios de n
def suma_divisores_propios(n):
    if n == 1:
        return 0 # El 1 no tiene divisores propios
    su = 1 # 1 siempre es divisor
    l = int(n**(1/2)) # Límite hasta la raíz cuadrada
    for i in range(2, l + 1):
        if n % i == 0:
            su += i # Añadir divisor
            if i != n // i:
                su += n // i # Añadir el divisor complementario
    return su

# Leer número límite
N = int(input())

# Imprimir suma de divisores para cada número
for i in range(1, N + 1):
    print(f"{i}: {suma_divisores_propios(i)}")
```