

A. Calcula la complejidad de las siguientes secuencias de código:

1.

```
a = 0
b = 0
for i in range(N):
    a = a + random()

for i in range(M):
    b = b + random()
```

2.

```
a = 0
for i in range(N):
    for j in reversed(range(i,N)):
        a = a + i + j
```

3.

```
k = 0
for i in range(n//2,n):
    for j in range(2,n,pow(2,j)):
        k = k + n / 2
```

4.

```
a = 0
i = N
while (i > 0):
    a += i
    i //= 2
```

5.

```
for i in range(n):
    while j<n:
        print("blank")
        j = j*k
```

6.

```
value = 0
for i in range(n):
    for j in range(i):
        value+=1
```

- B. Implementa un método .ordenar() para la clase Pila que ordene sus elementos (usando los métodos push() y pop())
- C. Define una clase ColaPrioridad para un hospital con 5 niveles de prioridad. Se dispone ya de la clase Cola y para la nueva clase se pide que se definan los atributos necesarios y los métodos queue() y dequeue() .
- D. Tu tarea es calcular el número de ceros a la derecha en el factorial n!.

Por ejemplo,

20! = 2432902008176640000 y tiene 4 ceros a la derecha.

Entrada:

La única línea de entrada contiene un entero n.

Salida:

Imprime el número de ceros finales en n!.

Restricciones:

$1 \leq n \leq 10^9$

Ejemplo:

Entrada:

20

Salida:

4

- E. Una permutación de los enteros 1, 2, ..., n se denomina “buena” si no hay elementos adyacentes cuya diferencia sea 1.

Dado n, construye una permutación buena si existe tal permutación.

Entrada:

La única línea de entrada contiene un entero n.

Salida:

Imprime una permutación hermosa de los enteros 1, 2, ..., n. Si hay varias soluciones, puedes imprimir cualquiera de ellas. Si no hay soluciones, imprime "NO SOLUTION".

Restricciones:

$$1 \leq n \leq 10^6$$

Ejemplo 1:

Entrada:

5

Salida:

4 2 5 3 1

Ejemplo 2:

Entrada:

3

Salida:

NO SOLUTION

- F. Hoy han empezado los mundiales, con victoria fácil para el equipo europeo. Sin embargo y debido a un error, los próximos días van a coincidir ciertas series en el calendario y será imposible ver todas. Tú y tu grupo de amigos queréis repartirlas para ver, entre todos, el máximo número posible.

Se jugarán n series y en tu grupo sois k amigos.

Conoces la hora de inicio y la hora de finalización de cada serie. ¿Cuál es el número máximo total de series que podéis ver por completo si actuáis de manera óptima?

Entrada:

La primera línea de entrada tiene dos enteros n y k : el número de series y el número de amigos.

Después de esto, hay n líneas que describen las series. Cada línea tiene dos enteros a y b : la hora de inicio y la hora de finalización de una serie.

Salida:

Imprime un solo entero: el número máximo total de películas.

Restricciones:

$$1 \leq n \leq 2 \times 10^5$$

$$1 \leq k \leq 100$$

$$1 \leq a < b \leq 10^9$$

Ejemplo:

Entrada:

5 2

1 5

8 10

3 6

2 5

6 9

Salida:

4