

Input, print and numbers

«Área de un triángulo rectángulo»

Enunciado

Escribe un programa que lea la longitud de la base y la altura de un triángulo rectángulo y calcule el área. Cada número se proporciona en una línea separada.

```
b = int(input())
h = int(input())
print((b*h)/2)
```

«Reparto de manzanas»

Enunciado

N estudiantes tienen K manzanas y las distribuyen equitativamente entre ellos. La parte restante (la que no se puede dividir) permanece en la cesta. ¿Cuántas manzanas recibe cada estudiante? ¿Cuántas manzanas quedan en la cesta?

El programa lee los números **N** y **K**. Debe imprimir las dos respuestas a las preguntas anteriores.

```
n = int(input())
k = int(input())
print(k // n)
print(k % n)
```

«El anterior y el siguiente»

Enunciado

Escribe un programa que lea un número entero e imprima el número anterior y el número siguiente. Observa los ejemplos a continuación para el formato exacto que deben tener las respuestas. No debe haber un espacio antes del punto final.

Recuerda que puedes convertir los números a cadenas de texto usando la función `str`.

```
a = int(input())
```

```
print("The next number for the number "+ str(a)+ " is "+ str(a+1))

print("The previous number for the number "+ str(a)+" is
"+str(a-1))
```

«Dos marcas de tiempo»

Enunciado

Una marca de tiempo se representa con tres números: horas, minutos y segundos. Dados dos momentos en el tiempo, calcula cuántos segundos hay entre ellos. El momento de la primera marca de tiempo ocurre antes que el de la segunda.

```
h1 = int(input())

m1 = int(input())

s1 = int(input())


h2 = int(input())

m2 = int(input())

s2 = int(input())


t1_en_segundos = h1 * 3600 + m1 * 60 + s1

t2_en_segundos = h2 * 3600 + m2 * 60 + s2


diferencia = t2_en_segundos - t1_en_segundos


print(diferencia)
```

«Pupitres escolares»

Enunciado

Una escuela decidió reemplazar los pupitres en tres aulas. Cada pupitre tiene capacidad para dos estudiantes. Dados los números de estudiantes en cada aula, imprime el menor número posible de pupitres que se pueden comprar.

El programa debe leer tres números enteros: el número de estudiantes en cada una de las tres clases, **a**, **b** y **c**, respectivamente.

En el primer caso de prueba, hay tres grupos. El primer grupo tiene 20 estudiantes y necesita 10 pupitres. El segundo grupo tiene 21 estudiantes, por lo que necesita al menos 11 pupitres. El tercer grupo tiene 22 estudiantes, y también necesita 11 pupitres. En total, se necesitan 32 pupitres.

```
a = int(input())
b = int(input())
c = int(input())

escritorios_a = (a+1) // 2
escritorios_b = (b+1) // 2
escritorios_c = (c+1) // 2

total_escritorios = escritorios_a + escritorios_b + escritorios_c

print(total_escritorios)
```

Integer and float numbers

«Último dígito»

Enunciado

Dado un número entero, imprime su último dígito.

```
a = int(input())  
print(a % 10)
```

«Dígito de las decenas»

Enunciado

Dado un número entero, imprime su dígito en la posición de las decenas.

```
n = int(input())  
  
print(n // 10 % 10)
```

«Suma de los dígitos»

Enunciado

Dado un número de tres dígitos, encuentra la suma de sus dígitos.

```
num = int(input())  
  
centenas = num // 100  
decenas = (num // 10) % 10  
unidades = num % 10  
  
sumaDigitos = centenas + decenas + unidades  
  
print(sumaDigitos)
```

«Parte fraccionaria»

Enunciado

Dado un número real positivo, imprime su parte fraccionaria.

```
num = float(input())

parteDecimal = num - int(num)

print(round(parteDecimal, 3))
```

«Primer dígito después del punto decimal»

Enunciado

Dado un número real positivo, imprime el primer dígito a la derecha del punto decimal.

```
num = float(input())

parteDecimal = num - int(num)

primerDigitoDecimal = int(parteDecimal * 10)

print(primerDigitoDecimal)
```

«Ruta del coche»

Enunciado

Un coche puede recorrer N kilómetros por día. ¿Cuántos días tardará en cubrir una ruta de M kilómetros? El programa recibe dos números: **N** y **M**.

```
import math

N = int(input())

M = int(input())

dias = math.ceil(M / N)
```

```
# Print the result

print(dias)
```

«Reloj digital»

Enunciado

Dado un número entero **N** - la cantidad de minutos que han pasado desde la medianoche - ¿cuántas horas y minutos muestra el reloj digital de 24 horas?

El programa debe imprimir dos números: el número de horas (entre 0 y 23) y el número de minutos (entre 0 y 59).

Por ejemplo, si **N = 150**, entonces han pasado 150 minutos desde la medianoche, es decir, ahora son las 2:30 am. Por lo tanto, el programa debe imprimir **2 30**.

```
N = int(input())

horas = N // 60

minutos = N % 60

print(horas, minutos)
```

«Costo total»

Enunciado

Un cupcake cuesta **A** dólares y **B** centavos. Determina cuántos dólares y centavos se deben pagar por **N** cupcakes. El programa recibe tres números: **A**, **B**, **N**. Debe imprimir dos números: el costo total en dólares y centavos.

```
euros = int(input())
centimos = int(input())
numeroCupcakes = int(input())

precio_por_cupcake_centimos = (euros * 100) + centimos
```

```
precio_total_centimos = precio_por_cupcake_centimos * numeroCupcakes

totalEuros = precio_total_centimos // 100

totalCentimos = precio_total_centimos % 100

print(totalEuros, totalCentimos)
```

Conditions: if, then, else

«Mínimo de tres números»

Enunciado

Dado tres números enteros, imprime el valor más pequeño.

```
if b >= a <= c:

    print(a)

elif a >= b <= c:

    print(b)

else:

    print(c)
```

«Números iguales»

Enunciado

Dado tres números enteros, determina cuántos de ellos son iguales entre sí. El programa debe imprimir uno de estos números:

- 3 (si los tres son iguales)
- 2 (si dos de ellos son iguales y el tercero es diferente)
- 0 (si todos los números son diferentes).

```
a = int(input())
b = int(input())
```

```

c = int(input())
if a == b == c:
    print(3)
elif a == b or b == c or a == c:
    print(2)
else:
    print(0)

```

«Movimiento de la torre»

Enunciado

La torre en ajedrez se mueve horizontal o verticalmente. Dadas dos casillas diferentes del tablero de ajedrez, determina si una torre puede ir de la primera casilla a la segunda en un solo movimiento.

El programa recibe como entrada cuatro números del 1 al 8: los dos primeros especifican la columna y la fila de la primera casilla, y los dos últimos la columna y la fila de la segunda casilla.

El programa debe imprimir **YES** si una torre puede ir de la primera casilla a la segunda en un movimiento, o **NO** en caso contrario.

```

x1 = int(input())
y1 = int(input())
x2 = int(input())
y2 = int(input())
if x1 == x2 or y1 == y2:
    print('YES')
else:
    print('NO')

```

«Tablero de ajedrez - mismo color»

Enunciado

Dadas dos casillas del tablero de ajedrez, si ambas tienen el mismo color, imprime la palabra **YES**; si tienen colores diferentes, imprime **NO**.

El programa recibe como entrada cuatro números del 1 al 8: los dos primeros especifican la

columna y la fila de la primera casilla, y los dos últimos la columna y la fila de la segunda casilla.

```
x1 = int(input())
y1 = int(input())
x2 = int(input())
y2 = int(input())
if (x1 + y1 + x2 + y2) % 2 == 0:
    print('YES')
else:
    print('NO')
```

«Movimiento del rey»

Enunciado

El rey en ajedrez se mueve horizontal, vertical o diagonalmente a cualquier casilla adyacente. Dadas dos casillas diferentes del tablero de ajedrez, determina si un rey puede ir de la primera casilla a la segunda en un solo movimiento.

El programa recibe como entrada cuatro números del 1 al 8: los dos primeros especifican la columna y la fila de la primera casilla, y los dos últimos la columna y la fila de la segunda casilla.

El programa debe imprimir **YES** si un rey puede ir de la primera casilla a la segunda en un movimiento, o **NO** en caso contrario.

```
x1 = int(input())
y1 = int(input())
x2 = int(input())
y2 = int(input())
if abs(x1 - x2) <= 1 and abs(y1 - y2) <= 1:
    print('YES')
else:
    print('NO')
```

«Movimiento del alfil»

Enunciado

En ajedrez, el alfil se mueve diagonalmente, cualquier cantidad de casillas. Dadas dos casillas diferentes del tablero de ajedrez, determina si un alfil puede ir de la primera casilla a la segunda en un solo movimiento.

El programa recibe como entrada cuatro números del 1 al 8, que especifican la columna y la fila de la casilla inicial y la columna y la fila de la casilla final.

El programa debe imprimir **YES** si un alfil puede ir de la primera casilla a la segunda en un movimiento, o **NO** en caso contrario.

```
x1 = int(input())
y1 = int(input())
x2 = int(input())
y2 = int(input())
if abs(x1 - x2) == abs(y1 - y2):
    print('YES')
else:
    print('NO')
```

«Movimiento de la reina»

Enunciado

La reina en ajedrez se mueve horizontal, vertical o diagonalmente cualquier cantidad de casillas. Dadas dos casillas diferentes del tablero de ajedrez, determina si una reina puede ir de la primera casilla a la segunda en un solo movimiento.

El programa recibe como entrada cuatro números del 1 al 8: los dos primeros especifican la columna y la fila de la primera casilla, y los dos últimos la columna y la fila de la segunda casilla.

El programa debe imprimir **YES** si una reina puede ir de la primera casilla a la segunda en un movimiento, o **NO** en caso contrario.

```
x1 = int(input())
y1 = int(input())
x2 = int(input())
y2 = int(input())
if abs(x1 - x2) == abs(y1 - y2):
    print('YES')
else:
    print('NO')
```

«Movimiento del caballo»

Enunciado

El caballo en ajedrez se mueve en forma de "L". Puede moverse dos casillas horizontalmente y una verticalmente, o dos casillas verticalmente y una horizontalmente.

Dadas dos casillas diferentes del tablero de ajedrez, determina si un caballo puede ir de la primera casilla a la segunda en un solo movimiento.

El programa recibe como entrada cuatro números del 1 al 8: los dos primeros especifican la columna y la fila de la primera casilla, y los dos últimos la columna y la fila de la segunda casilla.

El programa debe imprimir **YES** si un caballo puede ir de la primera casilla a la segunda en un movimiento, o **NO** en caso contrario.

```

x1 = int(input())
y1 = int(input())
x2 = int(input())
y2 = int(input())
dx = abs(x1 - x2)
dy = abs(y1 - y2)

if dx == 1 and dy == 2 or dx == 2 and dy == 1:
    print('YES')
else:
    print('NO')

```

«Barra de chocolate»

Enunciado

Una barra de chocolate tiene forma de un rectángulo dividido en $n \times m$ porciones. La barra de chocolate puede partirse en dos partes rectangulares rompiéndola a lo largo de una línea recta seleccionada.

Determina si es posible partirla de manera que una de las partes tenga exactamente k porciones.

El programa lee tres enteros: n , m y k . Debe imprimir YES o NO.

```

n = int(input())
m = int(input())
k = int(input())
if k < n * m and ((k % n == 0) or (k % m == 0)):
    print('YES')
else:
    print('NO')

```

«Año bisiesto»

Enunciado

Dado un año, verifica si es un año bisiesto. Si lo es, imprime **LEAP**; de lo contrario, imprime **COMMON**.

Las reglas del calendario gregoriano son las siguientes:

- Un año es bisiesto si es divisible exactamente entre 4, pero no divisible exactamente entre 100.
- Un año siempre es bisiesto si es divisible exactamente entre 400.

Advertencia: Las palabras **LEAP** y **COMMON** deben imprimirse en mayúsculas.

```
year = int(input())
if (year % 4 == 0) and (year % 100 != 0) or (year % 400 == 0):
    print('LEAP')
else:
    print('COMMON')
```

For loop with range

Given two integers A and B. Print all numbers from A to B inclusively, in ascending order, if $A < B$, or in descending order, if $A \geq B$.

```
a = int(input())
b = int(input())
if a < b:
    for i in range(a, b + 1):
        print(i)
else:
    for i in range(a, b - 1, -1):
        print(i)
```

En matemáticas, el factorial de un número entero n , denotado por $n!$, es el siguiente producto:

$$n! = 1 \times 2 \times \dots \times n$$

Para el número entero dado n , calcula el valor de $n!$. No utilices el módulo `math` en este ejercicio.

```
N = int(input())
total = 1

for i in range(1, N+1):
    total *= i
print(total)
```

Strings

Se te da una cadena de texto.

En la primera línea, imprime el tercer carácter de esta cadena.

En la segunda línea, imprime el penúltimo carácter de esta cadena.

En la tercera línea, imprime los primeros cinco caracteres de esta cadena.

En la cuarta línea, imprime todos los caracteres excepto los dos últimos de esta cadena.

En la quinta línea, imprime todos los caracteres de esta cadena con índices pares (recuerda que la indexación comienza desde 0, por lo que los caracteres se muestran comenzando con el primero).

En la sexta línea, imprime todos los caracteres de esta cadena con índices impares (es decir, comenzando con el segundo carácter de la cadena).

En la séptima línea, imprime todos los caracteres de la cadena en orden inverso.

En la octava línea, imprime cada segundo carácter de la cadena en orden inverso, comenzando desde el último.

En la novena línea, imprime la longitud de la cadena dada.

```
palabra = input()

print(palabra[2])
print(palabra[len(palabra)-2])
print(palabra[0:5])
print(palabra[0:len(palabra)-2])
print(palabra[::-2])
print(palabra[1::2])
print(palabra[::-1])
print(palabra[::-2])
```

```
print(len(palabra))
```

Given a string consisting of words separated by spaces. Determine how many words it has. To solve the problem, use the method `count`.

```
palabra = input()
print(palabra.count(" ") + 1)
```

REPLACE:

```
print(input().replace('1', 'one'))
```

While loop

Como futuro atleta, acabas de comenzar tu entrenamiento para un evento próximo. Dado que en el primer día corres x millas, y para el evento debes ser capaz de correr y millas, calcula el número de días necesarios para alcanzar finalmente la distancia requerida para el evento, si aumentas tu distancia cada día en un 10% respecto al día anterior.

Imprime un número entero que represente los días necesarios para alcanzar la distancia requerida.

```
x = int(input())
y = int(input())
dias = 1

while x < y:
    x = x + ((x*10) / 100)

    dias += 1
print(dias)
```