



503202/503203 Programación Programación Nivel Certamen en Python

EQUIPO PROGRAMACIÓN

4 de abril de 2025

Ejercicio de referencia

A. Ud. es parte de un grupo de amigos que se conocieron en la UdeC. Uno de sus amigos a tomado la valiente decisión de casarse y como se estila hace un tiempo, su amigo, el novio, ha contratado una lista de regalos de matrimonio en una de las tiendas de retail de la ciudad. La lista consta de diferentes tipos de regalo: cocinas, televisores, muebles, electrodomésticos, ropa de cama, etc. Para cada producto de la lista hay diferentes modelos y precios. Con su grupo de amigos (menos el novio, obviamente) han decidido poner una cuota y juntar un monto que les permita comprar uno de los modelos de cada producto, pero quieren saber cuánto deben poner cada uno si compran el modelo más barato de cada producto y cuánto deben poner si compran el modelo más caro de cada producto. Como saben que Ud. estudio ingeniería y que sabe programar (y Ud. no es el novio :D) le han pedido que haga un programa en Python que les permita resolver la situación.

Entrada: La entrada a este programa está compuesta por varias líneas. La primera contiene un número entero positivo menor o igual que 200, correspondiente a la cantidad de casos de prueba que ejecutará el programa. Cada caso de prueba estará compuesto por dos valores: un número entero positivo correspondiente a la cantidad de amigos que harán los regalos y un número C , ($1 \leq C \leq 20$), correspondiente a la cantidad de productos de la lista de novios en la tienda. Luego habrá C líneas, una por cada producto, cada una con varios números, el primero un valor M , ($1 \leq M \leq 20$), indicando la cantidad de modelos del producto y luego habrá M valores indicando el precio de cada modelo.

Salida: La salida a este programa estará compuesta por una línea por cada caso de prueba, cada línea contendrá dos números el valor de la cuota de cada amigo para comprar el modelo más barato de cada producto y el valor de la cuota de cada amigo para comprar el modelo más caro de cada producto.

Ejemplo de entrada 1:

```
2
5 3
3 800 600 400
2 500 1000
4 100 300 300 700
20 2
3 400 600 800
5 500 1000 300 100 900
```

Ejemplo de salida 1:

```
200 500
25 90
```

Solución:

```
def leeValida(m,M,txt):
    v=int(input(txt))
    while v<m or v>M:
        print(f'Error, el valor debe estar entre {m} y {M}')
        v=int(input(txt))
    return v

C=leeValida(1,200,'Casos de Prueba:')
for i in range(C):
    A=leeValida(1, float('inf'),'Cantidad de amigos:')
    C=leeValida(1,20,'Cantidad de productos de la lista:')
    suma_mayor=0
    suma_menor=0
    for j in range(C):
        M=leeValida(1,20,f'Cantidad de modelos del producto {j+1}:')
        mayor=0
        menor=float('inf')
        for k in range(M):
            P=leeValida(1,float('inf'),f'Precio del modelo {k+1} del producto {j+1}:')
            if P>mayor:
                mayor=P
            if P<menor:
                menor=P
        suma_mayor+=mayor
        suma_menor+=menor
    print(suma_menor/A, suma_mayor/A)
```

Ahora te toca a tí...

- 1.- Martín es un prodigioso estudiantes a quien las matemáticas le fascinan, siempre está pensando en operaciones raras con la ilusión de encontrar una relación numérica que se convierta en un nuevo descubrimiento.

Lo que en la actualidad le tiene inquieto es saber si existe alguna relación entre la cantidad de ceros y de otros dígitos presentes en el factorial de un número. Hasta ahora ha realizado experimentos con los números más pequeños, esto es:

```
0! es 1, uno tiene 0 ceros y un dígito distinto de cero.
1! es 1, uno tiene 0 ceros y un dígito distinto de cero.
2! es 2, dos tiene 0 ceros y un dígito distinto de cero.
3! es 6, seis tiene 0 ceros y un dígito distinto de cero.
4! es 24, veinticuatro tiene 0 ceros y dos dígitos distintos de cero.
etc.
```

A Martín le interesa tener un cuadro más completo de resultados de esta operación para establecer, si es que la hay, alguna relación.

Escriba un programa en Python que, dado un número entero calcule la cantidad de ceros y de dígitos distintos de cero de su factorial.

Entrada: La única entrada al programa es un número entero mayor o igual a cero.

Salidas: El programa tiene dos salidas, la cantidad de ceros y la cantidad de dígitos distintos de cero del factorial del número ingresado.

Ejemplo entrada 1: 4

Ejemplo salida 1: 0 y 2

Ejemplo entrada 2: 12

Ejemplo salida 2: 4 y 5

Observación: $12! = 479001600$ y este número tiene 4 ceros y 5 dígitos distintos de cero.

-
- 2.- Dados dos números enteros $N(2 \leq N \leq 1009)$ y $K(1 \leq K \leq 50)$ realizar K operaciones sucesivas bajo el siguiente algoritmo: si el último dígito de N es diferente de cero restarle 1 a N sino (el último dígito de N es cero) entonces dividir N por 10.

Entrada: La entrada al programa está compuesta por dos números enteros, N y K .

Salida: La única salida del programa es el valor final que toma N

Ejemplo de entrada: 512 4

Ejemplo de salida: 50

- 3.- Implemente un programa Python que calcule la operación potencia (x^y) usando sólo sumas. Los valores ingresados serán x e y , ambos números enteros.

Entradas: $x, x > 0$ e $y, y \geq 0$.

Salidas: La salida está compuesta por un único valor entero correspondiente a x^y .

Ejemplo de entrada 1: 6, 2.

Ejemplo de salida 1: 36.

Ejemplo de entrada 2: 3, 0.

Ejemplo de salida 2: 1.

Ejemplo de entrada 3: 5, 4.

Ejemplo de salida 3: 625.

- 4.- Para realizar una operación de multiplicación, los egipcios empleaban el siguiente método, para cuya explicación supondremos que se desea multiplicar 28×8 .

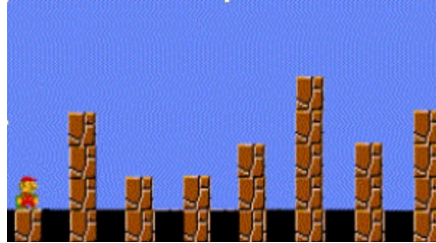
Se anota en 2 columnas los operandos	28	8
luego se registra la mitad de 28 y el doble de 8	14	16
luego se registra la mitad de 14 y el doble de 16	7	32
luego se registra la mitad de 7 (sin fracción) y el doble de 32	3	64
luego se registra la mitad de 3 y el doble de 64	1	128

Luego, se eliminan los números de la segunda columna que están frente a números pares de la primera columna y se suman los números de la segunda columna que no han sido eliminados obteniendo el resultado de la multiplicación:

Se elimina el 8 por estar delante de un 28 (par)	28	-
Se elimina el 16 por estar delante de un 14 (par)	14	-
se mantiene el 32	7	32
se mantiene el 64	3	64
se mantiene el 128	1	128
Resultado		224

Siguiendo este procedimiento, construya un programa Python que efectúe la multiplicación de 2 enteros usando el método de multiplicación Egipcia.

-
- 5.- Mario (Bros) ha llegado al castillo. Sólo le resta cruzar un conjunto de N murallas para ingresar a la recámara, vencer al monstruo y salvar a la princesa. En este problema estamos interesados en los saltos sobre las murallas. Para ello te indicaremos la altura de las N murallas recorridas desde la izquierda a la derecha. Mario se encuentra parado sobre la primera muralla y debe saltar a la siguiente muralla y así sucesivamente hasta llegar al otro extremo. Esto quiere decir que Mario dará $(N - 1)$ saltos. Un salto hacia arriba ocurrirá cuando Mario avance hacia una muralla más alta y, análogamente, un salto hacia abajo ocurrirá cuando Mario avance hacia una muralla más baja. El problema consiste en realizar un programa Python que determine cuántos saltos hacia arriba y cuántos saltos hacia abajo realizará Mario para ingresar al castillo.



Entradas : La primera línea de entrada contiene un entero N ($0 < N \leq 50$) que corresponde a la cantidad de murallas en el camino de Mario. La siguiente línea de entrada indicará la altura de cada una de las N murallas, desde la de la izquierda a la de la derecha. Cada altura es un entero positivo menor o igual a 10.

Salidas : El programa debe entregar dos números enteros, el primero corresponde a la cantidad de saltos hacia arriba y el segundo corresponde a los saltos hacia abajo.

Ejemplos de entradas 1: 8 1 4 2 2 3 5 3 4

Ejemplos de salida 1:

```
Saltos arriba : 4
Saltos abajo  : 2
```

Ejemplos de entradas 2: 5 1 2 3 4 5

Ejemplos de salida 2:

```
Saltos arriba : 4
Saltos abajo  : 0
```

6.- Abu Dhabi, la capital de los Emiratos Árabes Unidos, no es sólo exótica por sus lujosos hoteles o sus islas artificiales en las que han construido sus ostentosas mansiones muchos personajes famosos sino que los es porque en “Dhabi todo es posible”.

Así, en el “Abu Dabhi Equestrian Club”, uno de los casinos más grandes de los Emiratos, sus visitantes pueden jugar un peculiar dominó.



Primero, convengamos que en el juego de dominó convencional cada pieza, de forma rectangular, están divididas en dos secciones cuadradas en sus extremos, cada una de las cuales contiene entre cero y seis puntos. El juego completo contiene 28 piezas, las que contienen todas las combinaciones de puntos en sus extremos. No hay dos fichas iguales y no se hace distinción entre los extremos derecho e izquierdo, por tanto, la ficha 2-5 y la ficha 5-2 es la misma.

A diferencia del dominó ordinario, los puntos de las piezas del dominó en el Abu Dabhi Equestrian Club son diamantes reales de 10 quilates. Y no sólo eso, sino que cada extremo de una pieza puede tener entre cero y n diamantes. Entonces, la pregunta es ¿Cuántos diamantes se necesitan para producir un juego completo de este excéntrico dominó?. Obviamente depende del valor de n .

Entradas: La única entrada de este algoritmo es un número entero $n, 2 \leq n \leq 100$.

Salidas: La única salida de este algoritmo es un número entero correspondiente a la cantidad de diamantes requeridos para producir el juego completo.

Ejemplo de entrada 1: 2

Ejemplo de salida 1: 12

Ejemplo de entrada 2: 10

Ejemplo de salida 2: 660

-
- 7.- Simplificar una fracción es transformarla en una fracción equivalente más simple y para hacerlo dividimos su numerador y denominador por un mismo número. Por ejemplo, la fracción $\frac{12}{66}$ se puede transformar en la fracción $\frac{2}{11}$ dividiendo su numerador y denominador por 6. Desarrolle un algoritmo usando diagramas de flujo que lea el numerador y denominador de una fracción y que, de ser posible, despliegue una simplificación de ésta. Si la fracción no se puede simplificar el programa debe desplegar la frase 'Fracción irreducible'.

Entradas: La entrada al algoritmo consistirá en dos números enteros positivos n y d , correspondientes al numerador y denominador de una fracción, respectivamente.

Salidas: La salida del algoritmo dependerá de si la fracción se puede reducir o no. En el primer caso, el programa desplegará dos números enteros positivos correspondientes al numerador y denominador de la fracción de entrada simplificada. En el segundo caso debe desplegar el mensaje 'Fracción irreducible'.

Ejemplo de entrada 1: 12 66

Ejemplo de salida 1: 2 11 o también se aceptará 6 33 o 4 22

Ejemplo de entrada 2: 5 19

Ejemplo de salida 2: Fracción irreducible