



503202/503203 Programación Programación Básica en Python 4

EQUIPO PROGRAMACIÓN

31 de marzo de 2024

- 1.- Martín es un prodigioso estudiante 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.- 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.

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

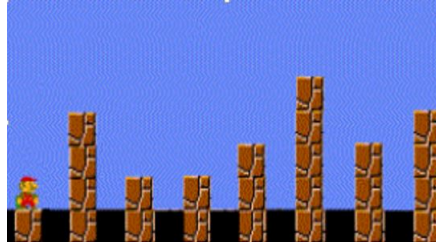
Se anota en 2 columnas los operandos	28	8
luego se registra la mitad de 28 y el doble de 8	14	16
luego se registra la mitad de 14 y el doble de 16	7	32
luego se registra la mitad de 7 (sin fracción) y el doble de 32	3	64
luego se registra la mitad de 3 y el doble de 64	1	128

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.

-
- 4.- 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
```

-
- 5.- 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

- 6.- 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