

Guía de Ejercicios 4

Introducción

En esta guía trabajaremos con ciclos en el control de flujo, en específico trabajaremos con el comando `while`.

Debajo de la mayoría de los problema encontrarás una tabla con ejemplos de entrada y salidas que deben producir tus soluciones. Usa estos ejemplos para verificar que tu solución sea correcta.

Ejercicios: `while`

1. Escribe un programa que pida dos números enteros al usuario. El programa debe imprimir el segundo número la cantidad de veces del primer número.

Entrada	Salida
4 2	2 2 2 2
3 0	0 0 0
0 10	

2. Escribe un programa que primero reciba un número entero, llamémoslo `n` . Seguido de esto deberá leer las siguientes `n` líneas, las cuales contendrán un número entero cada una. La salida del programa deberá ser la suma de todas esas líneas.

Entrada	Salida
3 4 2 1	7
2 3 -3	0
5 -1 -5 -7 -2 10	-5
0	

3. Escriba un programa que reciba un número n y uno m , ambos enteros positivos. Las siguientes n líneas serán enteros, los cuales deberá multiplicarlos entre sí. Si la multiplicación de estos números supera el valor de m , entonces deberá imprimir -1 , si no la supera entonces imprime la multiplicación total.

Entrada	Salida
4 7 4 2 -3 -1	-1
2 -4 -5 4	-20
3 28 -7 -2 2	28

4. Algo muy útil en programación competitiva es saber como encontrar el máximo y el mínimo entre un conjunto de números. Escriba un programa que reciba un número n , y luego recibirá

n números enteros positivos, todos entre 0 y 101. Luego tendremos dos casos. Si n es par, el programa debe retornar el número máximo de los n números recibidos. Si n es impar, entonces deberá retornar el mínimo del conjunto de números.

Entrada	Salida
4 4 2 3 1	4
3 4 2 1	1
5 7 2 60 20 30	2
6 17 40 100 4 8 30	100

5. Escriba un programa que reciba un número n y uno m , ambos enteros positivos. El programa deberá obtener el promedio de los siguientes n números y verificar si es mayor a m . Si es mayor el programa debe imprimir -1 . Si no lo es debe imprimir el promedio calculado previamente

Entrada	Salida
3 5 4 2 -3	1
5 2 3 1 1 4 2	-1
2 10 11 9	10

6. Escriba un programa que identifique si un número es primo. El programa recibe un número entero positivo de entrada. De salida retorna **Es primo** si el número de entrada lo es, o **No es primo** si es que no lo es.

Recordemos que un número es primo siempre y cuando los únicos números enteros positivos menores o iguales a él que generan resto 0 al dividirlo son 1 y el número en sí. Por ejemplo los números 2, 3, 5 y 7 son primos. El número 9 no es primo, ya que al ser dividido por 3 genera resto 0.

Entrada	Salida
7	Es primo
15	No es primo
81	No es primo
131071	Es primo
524287	Es primo
8388607	No es primo
2147483647	Es primo

7. Usted es un experto economista y matemático (supongamos que sí) y en su mundo solo existen billetes de 3 tipos, de 20, 50 y 100 dólares. Como es sabido mundialmente que usted puede resolver cualquier tipo de problema matemático económico, llegan a usted con el siguiente problema. Lo que quieren saber es que si lograrán para una cantidad exacta de billetes un monto exacto. Por ejemplo si le dicen que puede usar 10 billetes, y deben sumar 500 dolares, usted responde ¡Fácil! 10 billetes de 50 dólares (al instante). Ahora si le dicen que con 7 billetes debe sumar 600 dólares

¡Pan comido! 2 billetes de 50 y 5 de 100.

Ahora, usted aburrido de que le pregunten cosas tan fáciles decide escribir su algoritmo en un programa, de forma de que simplemente ejecuten el código y no le hagan perder su tiempo.

Escriba un programa que reciba la cantidad de billetes y el monto a sumar, y luego retorne la cantidad de billetes de cada tipo de forma que sumen la cantidad pedida. Ojo, que puede haber más de una respuesta correcta para los mismos datos de entrada.

Entrada	Salida
10 500	0 de 20, 10 de 50, 0 de 100
7 600	0 de 20, 2 de 50, 5 de 100
5 100	5 de 20, 0 de 50, 0 de 100
4 160	3 de 20, 0 de 50, 1 de 100
6 260	3 de 20, 2 de 50, 1 de 100

8. Ahora, las personas al darse cuenta de que les resolvió su problema con un simple código, esperan que les ponga a disposición la respuesta un problema más difícil en forma de código nuevamente. Lo que quieren saber ahora es si puede encontrar la cantidad mínima de billetes para cierto monto, más aún debe ver si es posible generar dicho monto con los 3 tipos de billete que dispone. Por ejemplo si le piden la cantidad mínima de billetes que sumen 120, son 1 billete de 100 y 1 de 20. Nótese que la misma suma puede obtenerse con 2 de 50 y 1 de 20 o con 6 de 20, pero en todas esas combinaciones se necesitan más billetes que en la respuesta correcta. Ahora si le piden con qué combinación de billetes puedo sumar 105 usted debe responder **No existe combinación de billetes que sume 110**.

Escriba un código que reciba un monto y responda la cantidad de cada billetes mínima de cada billete de forma que sumen ese monto. Si no es posible generar ese monto, debe responder que no se puede.

Entrada	Salida
500	0 de 20, 0 de 50, 5 de 100
620	1 de 20, 0 de 50, 6 de 100
105	No existe combinación de billetes que sume 105
830	4 de 20, 1 de 50, 7 de 100
270	1 de 20, 1 de 50, 2 de 100
308	No existe combinación de billetes que sume 308