

Ejercicios Segunda Evaluación – Fundamentos de Programación 22948 – 2023-1

1. Se trata de implementar un programa que sume los dígitos de un número entero no negativo. Por ejemplo, la suma de los dígitos del 3433 es 13.

El programa no debe limitarse a escribir el resultado de la suma, sino que también escribirá todos los sumandos utilizados: $3 + 4 + 3 + 3 = 13$.

La entrada consta de una serie de casos de prueba terminados con un número negativo. Cada caso de prueba es una simple línea con un número no negativo no mayor que 10^9 del que habrá que sumar todos sus dígitos.

Para cada caso de prueba se mostrará una línea en la que aparecerán cada uno de los dígitos separados por el signo +, tras lo cual aparecerá el símbolo = y la suma de todos los dígitos.

Tenga en cuenta que antes y después de cada símbolo (+ y =) hay un espacio.

Ejemplos de entradas y salidas:

3433	3 + 4 + 3 + 3 = 13
64323	6 + 4 + 3 + 2 + 3 = 18
8	8 = 8
-1	

2. Dado un $N \geq 2$, se llaman números afortunados a los que resultan de ejecutar el siguiente proceso: se comienza generando una lista que contiene los números desde 1 hasta N en este orden; se elimina de la lista un número de cada 2 (es decir, los números 1, 3, 5, etc.); de la lista final resultante se elimina un número de cada 3, etc. El proceso termina cuando se va a eliminar un número de cada M y el número de elementos que quedan es menor que M. Los números que queden en la lista en este momento son los afortunados.

La entrada consistirá en distintos casos de prueba, cada uno en una línea. Cada caso de prueba contiene el número con el que se calcularán los números afortunados. Ese número será siempre mayor o igual que 2 y no mayor que 100.000. La entrada terminará con un 0, que marcará el final de la entrada y no generará salida.

Para cada caso de prueba se escribirá una línea que contendrá al principio el N de partida, seguido por dos puntos, un espacio y todos los números afortunados en orden decreciente.

Ejemplos de entradas y salidas:

3	3: 2
10	10: 10 6 4
30	30: 30 22 18 12 10
0	

3. Dado el valor de una base x ($1 \leq x \leq 100$) y el valor de un exponente n ($0 \leq n \leq 1.000.000$), se trata de realizar el siguiente cálculo:

$$x^0 + x^1 + x^2 + \dots + x^n$$

La entrada estará compuesta por distintos casos de prueba. Cada uno de ellos aparecerá en una línea independiente que tendrá dos números enteros x y n .

Para cada caso de prueba se escribirá, en una línea independiente, el resultado de evaluar la serie de potencias anterior para los valores de x y n dados.

Para evitar problemas de desbordamientos, se escribirá el resultado módulo 1.000.007.

Ejemplos de entradas y salidas:

1 2	3
2 3	15
3 4	121
2 19	48568

4. Escribir un programa que pida números enteros positivos y los convierta en romanos. El programa debe pedir números enteros hasta que se introduzca un 0, lo que detiene el ciclo.

5. Construir un programa que lea un número entero positivo n y después genere n valores de manera aleatoria (entre 0 y 10) y los guarde en un vector. El programa deberá hallar la media de los elementos, y después buscar entre ellos aquellos que sean mayores que la media. El resultado que entregara el programa es la media de los números y una lista de elementos que cumplan con la condición de ser mayores que la media. (nota: se deberá usar un segundo vector para la lista de elementos mayores que la media). Una vez obtenido el vector de elementos mayores que la media, ordenarlos de menor a mayor, e imprimir la lista ordenada.

6. Realizar el problema presentado en <https://acceptaelreto.com/problem/statement.php?id=111&cat=13>

7. Realizar el problema presentado en <https://acceptaelreto.com/problem/statement.php?id=123&cat=13>

8. Construir un programa que reste dos fechas, obteniendo el número de días entre ellas. Hallar el resultado teniendo en cuenta si hay años bisiestos entre ellas.

Ejercicios tomados de:

<https://acceptaelreto.com/>, una página web creada por un grupo de profesores de la Universidad Complutense de Madrid.

Programación en C, de B. Gottfried.