**EN CADA UNO DE LOS PROGRAMAS, SE DEBE INCLUIR VALIDACION DE RANGO DE DATOS SEGÚN CORRESPONDA. ADEMAS, SE REQUIEREN SOLUCIONES MODULARES!**

1. Escribe un programa Java para verificar si un número dado es un número feo.

*En el sistema numérico, los números feos son números positivos cuyos únicos factores primos son 2, 3 o 5. Los primeros 10 números feos son 1, 2, 3, 4, 5, 6, 8, 9, 10, 12. Por convención, 1 es incluido.*

Entrada Ejemplo: Ingrese un número entero: 235

Salida Esperada: No es un número feo.

2. Escriba un programa Java para determinar la cantidad de números Abundantes, Deficiente, y Perfectos (enteros) entre 1 y 10,000.

* En teoría de números, un número abundante es un número para el cual la suma de sus divisores apropiados es mayor que el número mismo. Por ejemplo, los primeros números abundantes son: 12, 18, 20, 24, 30, 36, 40, 42, 48, 54, 56, 60, 66, 70, 72, 78, 80, 84, 88, 90, 96, 100, 102,… Así, el número entero 12 es el primer número abundante. Sus divisores adecuados son 1, 2, 3, 4 y 6 para un total de 16.
* Número deficiente: en teoría de números, un número deficiente es un número n para el cual la suma de los divisores σ(n) < 2n. El valor 2n - σ (n) se llama deficiencia del número. Como ejemplo, los divisores de 21 son 1, 3 y 7, y su suma es 11. Como 11 es menor que 21, el número 21 es deficiente. Su deficiencia es 2 × 21 - 32 = 10. Los primeros números deficientes son: 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 21, 22, 23, 25, 26, 27, 29, 31, 32, 33, …….
* Número perfecto: en el sistema numérico, un número perfecto es un entero positivo que es igual a la suma de sus divisores positivos propios, es decir, la suma de sus divisores positivos excluyendo el número en sí. De manera equivalente, un número perfecto es un número que es la mitad de la suma de todos sus divisores positivos (incluido él mismo), es decir, σ1 (n) = 2n. El primer número perfecto es 6. Sus divisores propios son 1, 2 y 3, y 1 + 2 + 3 = 6. De manera equivalente, el número 6 es igual a la mitad de la suma de todos sus divisores positivos: (1 + 2 + 3 + 6) / 2 = 6. El siguiente número perfecto es 28 = 1 + 2 + 4 + 7 + 14. A esto le siguen los números perfectos 496 y 8128.

Salida Esperada:

Recuento de números [(enteros) entre 1 y 10,000]:

Número abundante: 2488

Número deficiente: 7508

Número perfecto: 4

3. Escriba un programa Java para dado un número, determinar si este es Perfecto, Abundante o Deficiente.

* En teoría de números, un número abundante es un número para el cual la suma de sus divisores apropiados es mayor que el número mismo. Por ejemplo, los primeros números abundantes son: 12, 18, 20, 24, 30, 36, 40, 42, 48, 54, 56, 60, 66, 70, 72, 78, 80, 84, 88, 90, 96, 100, 102,… Así, el número entero 12 es el primer número abundante. Sus divisores adecuados son 1, 2, 3, 4 y 6 para un total de 16.
* Número deficiente: en teoría de números, un número deficiente es un número n para el cual la suma de los divisores σ(n) < 2n. El valor 2n - σ (n) se llama deficiencia del número. Como ejemplo, los divisores de 21 son 1, 3 y 7, y su suma es 11. Como 11 es menor que 21, el número 21 es deficiente. Su deficiencia es 2 × 21 - 32 = 10. Los primeros números deficientes son: 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 21, 22, 23, 25, 26, 27, 29, 31, 32, 33, …….
* Número perfecto: en el sistema numérico, un número perfecto es un entero positivo que es igual a la suma de sus divisores positivos propios, es decir, la suma de sus divisores positivos excluyendo el número en sí. De manera equivalente, un número perfecto es un número que es la mitad de la suma de todos sus divisores positivos (incluido él mismo), es decir, σ1 (n) = 2n. El primer número perfecto es 6. Sus divisores propios son 1, 2 y 3, y 1 + 2 + 3 = 6. De manera equivalente, el número 6 es igual a la mitad de la suma de todos sus divisores positivos: (1 + 2 + 3 + 6) / 2 = 6. El siguiente número perfecto es 28 = 1 + 2 + 4 + 7 + 14. A esto le siguen los números perfectos 496 y 8128.

Entrada Ejemplo: 6

Salida Esperada:

6 No es un Número abundante

6 No es un Número deficiente

6 No es un Número perfecto

4. Escriba un programa Java para generar y mostrar enteros aleatorios en un rango específico.

5. Escriba un programa Java para generar y mostrar todos los números de Kaprekar menores que un número entero definido, además de mostrar el total de estos números.

*En matemáticas, un número de Kaprekar es aquel entero no negativo tal que, en una base dada, los dígitos de su cuadrado en esa base pueden ser separados en dos números que sumados dan el número original.*

Entrada Ejemplo: 1000

Salida Esperada:

1 1 0 + 1

9 81 8 + 1

45 2025 20 + 25

55 3025 30 + 25

99 9801 98 + 01

297 88209 88 + 209

703 494209 494 + 209

999 998001 998 + 001

8 números de Kaprekar.

6. Escriba un programa Java para generar y mostrar los primeros N números decimales narcisistas, donde N es un número positivo dado por el usuario.

*Un número narcisista​ es aquel que es igual a la suma de sus dígitos elevados a la potencia de su número de cifras.*

Entrada Ejemplo: 15

Salida Esperada:

0 1 2 3 4 5 6 7 8 9 153 370 371 407 1634

7. Escriba un programa Java para mostrar los primeros N números de Lucus, donde N es un número positivo dado por el usuario.

*Los números o series de Lucas son una secuencia entera nombrada en honor al matemático François Édouard Anatole Lucas, quien estudió tanto esa secuencia como los números de Fibonacci estrechamente relacionados. Los números de Lucas y los números de Fibonacci forman instancias complementarias de las secuencias de Lucas. La secuencia de los números de Lucas es: 2, 1, 3, 4, 7, 11, 18, 29*,….

Entrada Ejemplo: 10

Salida Esperada:

Los primeros diez números de Lucas a:

2

1

3

4

7

11

18

29

47

76

8. Escribe un programa Java para imprimir los primeros N números catalanes, donde N es un número positivo dado por el usuario.

*En combinatorias, los números catalanes forman una secuencia de números naturales que ocurren en varios problemas de conteo, que a menudo involucran objetos definidos recursivamente. Llevan el nombre del matemático belga Eugène Charles Catalan. El n-ésimo número de Catalan se obtiene, aplicando coeficientes binomiales, a partir de la siguiente fórmula:*

**

*{\displaystyle C\_{n}={\frac {1}{n+1}}{2n \choose n}={\frac {(2n)!}{(n+1)!\,n!}}\qquad {\mbox{ con }}n\geq 0.}Así, los primeros números catalanes para n = 0, 1, 2, 3,… son 1, 1, 2, 5, 14, 42, 132, 429, 1430, 4862, 16796, 58786, 208012, 742900, 2674440, 9694845, 35357670 , 129644790, 477638700, 1767263190, 6564120420, 24466267020, 91482563640, 343059613650, 1289904147324, 4861946401452,*

Entrada Ejemplo: 10

Salida Esperada:

Lista 10 números catalanes: -

1

2

5

14

42

132

429

1430

4862

16796

9. Escribe un programa Java para encontrar e imprimir los primeros N números felices, donde N es un número positivo dado por el usuario.

*Número feliz: Comenzando con cualquier entero positivo, reemplaza el número por la suma de los cuadrados de sus dígitos y repite el proceso hasta que el número sea igual a 1, o se repite sin fin en un ciclo que no incluye 1. Ejemplo: 19 es un número feliz*

*12 + 92 = 82*

*82 + 22 = 68*

*62 + 82 = 100*

*12 + 02 + 02 = 1*

Entrada Ejemplo: 10

Salida Esperada:

Los primeros 10 números felices:

1

7

10

13

19

23

28

31

10. Escriba un programa Java para verificar si un número dado es un número feliz o un número infeliz.

* *Número feliz: Comenzando con cualquier entero positivo, reemplaza el número por la suma de los cuadrados de sus dígitos y repite el proceso hasta que el número sea igual a 1, o se repite sin fin en un ciclo que no incluye 1.*
* *Un número infeliz es un número que no es feliz. Los primeros números infelices son 2, 3, 4, 5, 6, 8, 9, 11, 12, 14, 15, 16, 17, 18, 20.*

Entrada Ejemplo: 5

Salida Esperada: Número infeliz

11. Escriba un programa Java para verificar si un número dado es un número de Disarium o un número infeliz.

*Un número es disarium si la suma de sus dígitos a la potencia de su posición es igual al mismo número. Por ejemplo 135 es un número de Disariumm porque 11 + 32 + 5*3 = 135. *Algunos otros DISARIUM son 89, 175, 518 etc.*

Entrada Ejemplo: 25

Salida Esperada: No es un número de disario.

12. Escriba un programa Java para verificar si un número es un Número de Harshad o no.

*En matemáticas, un número de Harshad es un número entero que es divisible por la suma de sus dígitos. Ejemplo: El número 200 es un número de Harshad porque la suma de los dígitos 2 y 0 y 0 es 2 (2 + 0 + 0) y 200 es divisible por 2. El número 171 es un número de Harshad porque la suma de los dígitos 1 y 7 y 1 es 9 (1 + 7 + 1) y 171 es divisible por 9.*

Entrada Ejemplo: 353

Salida Esperada:  353 no es un número de Harshad.

13. Escriba un programa Java para verificar si un número es un Número Prónico o no.

*Un número Prónico es un número que es el producto de dos enteros consecutivos, es decir, un número de la forma n (n + 1). Los primeros números Prónicos son: 0, 2, 6, 12, 20, 30, 42, 56, 72, 90, 110, 132, 156, 182, 210, 240, 272, 306, 342, 380, 420, 462 ... etc.*

Entrada Ejemplo: 110

Salida Esperada:Número Prónico.

14. Escriba un programa de Java para verificar si un número es un número Automórfico o no.

*En matemáticas, un número Automórfico es un número cuyo cuadrado "termina" en los mismos dígitos que el número mismo. Por ejemplo, 52 = 25, 62 = 36, 762 = 5776 y 8906252 = 793212890625, por lo que 5, 6, 76 y 890625 son todos números Automórficos.*

Entrada Ejemplo: 76

Salida Esperada: Número Automórfico.

15. Escriba un programa Java para verificar si un número es un Número de Pato o no.

*Un número de Pato es un número que tiene ceros presentes, pero no debe haber cero, presente al principio del número. Por ejemplo, 3210, 7056, 8430709 son todos los números de Pato, mientras que 08237, 04309 no lo son.*

Entrada Ejemplo: 3210

Salida Esperada: Número de pato

16. Escriba un programa Java para verificar que dos números sean números amistosos o no.

*Los números amistosos son dos números diferentes tan relacionados que la suma de los divisores de cada uno es igual al otro número. Por ejemplo, los primeros diez pares amistosos son: (220, 284), (1184, 1210), (2620, 2924), (5020, 5564), (6232, 6368), (10744, 10856), (12285, 14595), ( 17296, 18416), (63020, 76084) y (66928, 66992).*

Entrada Ejemplo:

220

228

Salida Esperada: Estos números son amistosos.

17. Escriba un programa Java para verificar si un número dado es circular primo o no.

*Número circular primo: Un número circular primo tiene la propiedad de que el número generado cuando se permutan cíclicamente sus dígitos, dichos números son primos. Por ejemplo, 1193 es un número circular primo, ya que 1931, 9311 y 3119 también son primos. Un número circular primo con al menos dos dígitos solo puede consistir en combinaciones de los dígitos 1, 3, 7 o 9, porque tener 0, 2, 4, 6 u 8 como el último dígito hace que el número sea divisible por 2.*

Entrada Ejemplo: 35

Salida Esperada: No es un número circular prime.

18. Escribir un programa Java para verificar un número es un cubo o no.

*En aritmética y álgebra, el cubo de un número n es su tercera potencia, es decir, el resultado del número multiplicado por sí mismo dos veces: n3 = n × n × n.*

Entrada Ejemplo: 8

Salida Esperada: El número es un cubo.

19. Escribir un programa Java para verificar si un número es cíclico o no.

*Un número cíclico es un número entero en el que las permutaciones cíclicas de los dígitos son múltiplos sucesivos del número. Un conocido ejemplo es 142857:*

*142857 × 1 = 142857*

*142857 × 2 = 285714*

*142857 × 3 = 428571*

*142857 × 4 = 571428*

*142857 × 5 = 714285*

*142857 × 6 = 857142*

Entrada Ejemplo: 142857

Salida Esperada: Es un número cíclico.

20. Escriba un programa Java para mostrar los primeros N números Fermat, donde N es un número positivo ingresado por el usuario.

*En matemáticas, un número Fermat es un entero positivo de la forma*Fermat Number *donde n es un entero no negativo. Los primeros números de Fermat son:*

*3, 5, 17, 257, 65537, 4294967297, 18446744073709551617,…*

Entrada Ejemplo: 5

Salida Esperada:

3.0

5.0

17.0

257.0

65537.0

21. Escriba un programa Java para determinar si un número se puede expresarse como la suma de dos cubos en al menos dos (o más) formas diferentes.

 Aquí hay algunos ejemplos:

1729 = 1^3 + 12^3 = 9^3 + 10^3  
1729 = 1^3 + 12^3 = 9^3 + 10^3  
4104 = 2^3 + 16^3 = 9^3 + 15^3  
1729 = 1^3 + 12^3 = 9^3 + 10^3  
4104 = 2^3 + 16^3 = 9^3 + 15^3  
13832 = 2^3 + 24^3 = 18^3 + 20^3  
39312 = 2^3 + 34^3 = 15^3 + 33^3  
46683 = 3^3 + 36^3 = 27^3 + 30^3  
32832 = 4^3 + 32^3 = 18^3 + 30^3

Entrada Ejemplo: 40033

*Salida Esperada:* 9^3 + 34^3 = 16^3 + 33^3

22. Escriba un programa para verificar si un número es el número de Mersenne o no.

*En matemáticas, un número de Mersenne es un número primo que se puede escribir en la forma Mn = 2n - 1 para algún entero n. Los primeros cuatro números primos de Mersenne son 3, 7, 31 y 127*

Entrada Ejemplo: 127

*S*alida Esperada: 127 es un número de Mersenne.

23. Escriba un programa Java para encontrar todos los números narcisistas entre 1 y N, donde N es un número positivo que ingresa el usuario.

*En teoría de números, un número narcisista es un número que es la suma de sus propios dígitos, cada uno elevado a la potencia del número de dígitos. Por ejemplo: 153 = 13 + 53 + 33*

Entrada Ejemplo: 10

Salida Esperada:

1

2

3

4

5

6

7

8

9

153

24. Escriba un programa Java para verificar si un número es palíndromo o no.

*En el sistema numérico, un número palíndromo es un número que es el mismo cuando se escribe hacia adelante o hacia atrás. Los primeros números palíndromos son 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 22, 33, 44, 55, 66, 77, 88, 99, 101, 111, ...*

Entrada Ejemplo: 5

Salida Esperada: Es un número de palíndromo.

25. Escribe un programa Java para imprimir los primeros N números de la serie Pell, donde N es un número positivo que ingresa el usuario.

*En matemáticas, los números de Pell son una secuencia infinita de enteros. La secuencia de los números de Pell comienza con 0 y 1, y luego cada número de Pell es la suma de dos veces el número de Pell anterior y el número de Pell anterior a ese. Así, los primeros términos de la secuencia son:0, 1, 2, 5, 12, 29, 70, 169, 408, 985, 2378, 5741, 13860,…*

Entrada Ejemplo: 20

Salida Esperada: 1 2 5 12 29 70 169 408 985 2378 5741 13860 33461 80782 195025 470832 113

6689 2744210 6625109 15994428

26. Escriba un Programa en Java para verificar si un número es un Número de Keith o no.

*Para determinar si un número entero positivo N, de n cifras, es un número de Keith, se crea una sucesión entera. Esta sucesión tiene como primeros términos los n dígitos decimales de N, ordenándose desde el dígito más significativo. Se continúa la sucesión, donde cada término es la suma de los anteriores n números. Por definición, N es un número de Keith si N aparece en la secuencia construida. Por ejemplo, considerar el número de 3 dígitos N = 197. La secuencia por construir entonces es:* ***1****,****9****,****7****, 17, 33, 57, 107, 197, 361, ...Se observa que 197 se encuentra en la sucesión. Por tanto, se afirma que 197 es un****número de Keith.***

Entrada Ejemplo: 75

Salida Esperada: Número de Keith

27. Escribe un programa Java para crear los primeros N números de Hamming, donde N es un número positivo que ingresa el usuario.

*En informática, los números regulares a menudo se llaman números de Hamming, los números de Hamming son números cuyos únicos factores primos son 2, 3 y 5. Es decir, los números de Hamming forman una sucesión estrictamente creciente de números que cumplen las siguientes condiciones:*

1. *El número 1 está en la sucesión.*
2. *Si x está en la sucesión, entonces 2x, 3x y 5x también están.*
3. *Ningún otro número está en la sucesión.*

*Los primeros números de Hamming son: 1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 25, 27, 30, 32*

Entrada Ejemplo: 20

Salida Esperada: Primeros veinte números de Hamming: 1 2 3 4 5 6 8 9 10 12 15 16 18 20 24 25 27 30 32 36