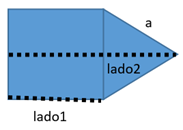
**TALLER # 1**

**PROGRAMACIÓN 1**

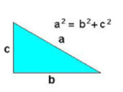
Nota: Para resolver este taller debe hacer uso de las funciones leerEntero, leerDouble, leer String e imprimir presentes al final de este taller. Por favor no mezclar lógica con interfaz.

Crear una aplicación para:

1. Incrementar el salario de un empleado en un 10%
2. Informar el salario previo, si se sabe el porcentaje de incremento del salario y el nuevo salario
3. Informar cuál fue el porcentaje de incremento, si se sabe el salario incrementado y el salario anterior
4. calcular el lado de un tetraedro regular, si se sabe el volumen.
5. calcular el radio, si se sabe el volumen de una esfera. Recuerde que volumenEsfera = 4/3 𝞹 r3
6. Una finca tiene la siguiente forma:



hallar el perímetro de toda la finca y el área, si se conocen las longitudes que se encuentran punteadas.



Tenga en cuenta que debe usar Pitágoras.

1. Un coleccionista de antigüedades tiene un reloj que se mueve en sentido inverso (tal como sería al ver un reloj normal mediante un espejo) . El coleccionista tiene algunas dificultades en saber la hora correcta, por lo tanto necesita un algoritmo (**pseudocódigo**) que dada la hora y los minutos del reloj, imprima la hora correcta. Puede usar la función **valorAbsoluto**.

**Ejemplo:** Suponga que en el reloj del espejo son las 7 horas con 10 minutos, el programa debe indicar que la hora correcta es 5:50.

**public** **static** **int** calcularHoraReal ( **int** horaReloj )

{

}

**public** **static** **int** calcularMinutoReal ( **int** minutoReloj )

{**int** minReal=0;

**return** minReal;

}

**public** **static** String devolverHoraReal(**int** minutos, **int** horaReloj)

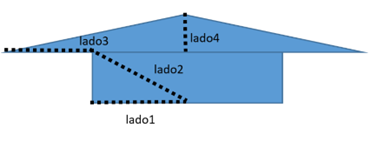
{

String resultado="";

//invocar calcularHoraReal y calcularMinutoReal

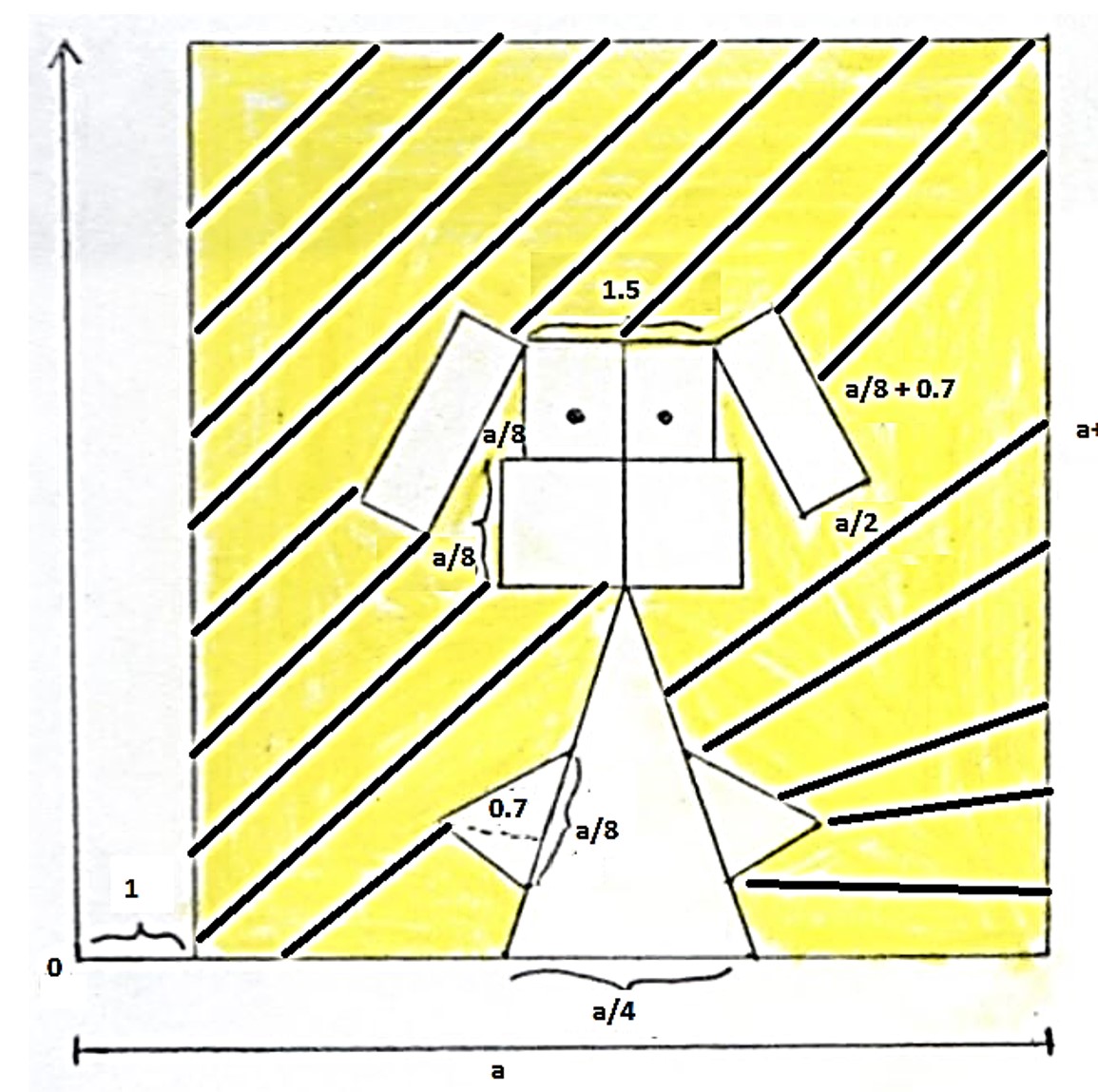
**return** resultado;}

1. Calcular el área de la casa si se conocen las longitudes punteadas



| Variables | Expresión área de la casa |
| --- | --- |
| Entrada:  Salida: |  |

1. Construir un programa que calcule el área sombreada (rayada) del rectángulo, si se ha dibujado una mascota dentro de él[1].

****

**public** **class** Mascota {

**public** **static** **double** calcularAreaTriangulo(**double** base, **double** altura)

{

**double** area= (base\*altura)/2;

**return** area;

}

**public** **static** **double** calcularAreaRectangulo(**double** largo, **double** ancho)

{

**double** area= (largo\*ancho);

**return** area;

}

**public** **static** **double** calcularAreaCuadrado(**double** lado)

{

**double** area= (lado\*lado);

**return** area;

}

**public double calcularAreaCabeza (double a)**

**{//aqui debe invocar los métodos para calcularAreas**

**double areaCabeza=0;**

**return areaCabeza;**

**}**

**public double calcularAreaCuerpo (double a)**

**{//aqui debe invocar los métodos para calcularAreas**

**double areaCuerpo=0;**

**return areaCuerpo; }**

**public double calcularAreaTotalMascota(double a)**

**{**

**}**

**public double calcularAreaSombreada(double a)**

**{**

**}**

**public static void main(String[] args) {**

**}**

**}**

[1] Figura basada en http://compumch.webnode.es/a1%C2%B0-primaria-iii-b/actividad-3-19-8-15-herramienta-borrador-dibujo-figuras-geometricas/

1. Construir un problema que calcule las raíces de una ecuación cuadrática



1. Informa el lugar que visitará un turista, si se sabe el mes en el que desea viajar, la edad del turista, la cantidad de días a permanecer en el territorio, y su presupuesto. El sitio asignado para el viaje depende de las estaciones, de la disponibilidad para trabajar y de otras condiciones que se presentan a continuación:

-Si el mes en el que el turista desea viajar corresponde a Invierno, hay dos opciones de viaje: Turkia o Gran Bretaña . Si su presupuesto es superior a 20 millones, el turista viajará a Gran Bretaña. Cuando el turista sea menor de edad, sin importar su presupuesto, viajará a Turkia.

- Cuando el mes corresponda a Verano, el turista, que debe ser mayor de edad, viajará a Dubai, siempre y cuando permanezca más de 10 días en el territorio y pueda trabajar como mínimo 5 días en Uber. De lo contrario irá a Cancún.

- Si el mes de viaje corresponde a Otoño, el turista deberá tener edad entre 40 y 65 años, y un presupuesto de 10 millones, viajará a Chile.

- Si el mes es Primavera, y el turista no puede trabajar, pero su presupuesto es superior a 22 millones, irá a Japón.

- Si el turista no cumple con ninguna condición previa, deberá viajar a Perú.

Informar el sitio que debe visitar el turista.

Tabla de equivalencia Mes - estación.

12, 1, 2 Invierno ; 3, 4, 5 Primavera ; 6, 7, 8 Verano ; 9, 10, 11 Otoño

1. Se desea crear una aplicación para informar el obsequio que recibirá una mujer en una tienda de ropa. Las condiciones para la entrega del obsequio son las siguientes:

- Si tiene tarjeta Platinum, un acumulado de puntos superior a 20000, sus compras anuales son inferiores a 4.5 millones y tiene la tarjeta desde hace 2 o más años, se entregará un bolso.

- Si su tarjeta es platinum, el acumulado de puntos es inferior o igual a 20000 recibirá una cosmetiquera.

- Si su tarjeta de puntos es Gold, posee la tarjeta desde hace 2 o más años, tiene edad superior a 45 años y acumulado de compras anuales superior a 10 millones, se le dará una pulsera.

- Si su tarjeta es Gold, su edad está entre 18 y 45 años y cumple años en el mes actual, se le regalará un Brazalete.

- Si tiene alguna de las tarjetas, y no cumplió ninguna de las condiciones anteriores, y su promedio de compras anuales es de 6 millones de pesos se le regalará una cosmetiquera.

- Si no tiene tarjeta pero su promedio de compras excede los 15 millones, se le regalará iphone 6s.

- Si no cumple ninguna de las condiciones anteriores no recibirá obsequio.

1. Cree un programa que permita leer un número de 3 dígitos ( un solo entero) y calcule cuántos de esos dígitos son números primos
2. Una empresa de encomiendas necesita un programa que permita calcular el precio de un envío de acuerdo a la siguiente información:
3. Existen 3 tipos de envío: normal, rápido, inmediato, los cuales tienen un costo respectivo de 1000, 1500 y 2000 pesos por cada gramo del peso del paquete.
4. Si el producto pesa de 10 a 30 kilos, el precio del envío tendrá un incremento del 15%, pero si pesa entre 31 y 60 kilos, tendrá un incremento del 20%, si pesa más de 60 kilos tendrá un incremento de 25% más 50 pesos por cada kilo de mas (ejemplo si pesa 61 tiene un kilo de más)
5. Cree un algoritmo que dada una fecha (dd , mm, anio) permita sumarle un número entero de días. Tenga en cuenta que al sumarle los días hay que verificar si hay cambio de mes, o de año, así mismo si se trata de una año bisiesto( febrero es de 28 0 29 días).
6. Se dice que un número es *hyperpar* cuando todos sus dígitos son pares.  
    Por ejemplo: 2460 cada dígito 2, 4, 6, 0 son pares.  
    Crear una función que reciba un número no negativo entre 0 y 9999 y diga si el número es hyperpar o no.

public static boolean esHyperpar(int numero) {

boolean bandera=false;

//Aquí va el código de la función.

return bandera

}

1. Se denomina número reversible a aquél que al ser sumado a sí mismo tras invertir sus dígitos da como resultado un número en el que todos los dígitos son impares.  
   Por ejemplo, el número 36 es reversible pues 36 + 63 = 99, y los dos dígitos de 99 son impares. También lo son el 409 y el 904  
   Crea una función que reciba un número entre 0 y 999 y retorne como salida *true* si el número es reversible o *false* si no lo es.

public static boolean esReversible( int numero) {

boolean bandera = false;

// Aquí va el cuerpo de la función

return bandera;

}

1. Corrección de errores, En las transmisiones de radio se emplea un sistema de corrección de errores, el cual funciona recibiendo varias versiones del mensaje (3 veces) y componiendo el mensaje según los datos recibidos, para verificar que el dato es correcto, debe recibirse al menos dos veces el dato. Ejemplo:

| **Dato1** | **Dato2** | **Dato 3** | **Mensaje Correcto** |
| --- | --- | --- | --- |
| 123 | 123 | 123 | 123 |
| 124 | 113 | 23 | 123 |
| 121 | 190 | 305 | Error de transmisión |
| 208 | 303 | 218 | 208 |

Crea una función que reciba tres números entre 0 y 999 correspondientes a mensajes recibidos e identifique cuál es el mensaje correcto o indique si hay error de transmisión.

public static void correccionErrores(int a, int b, int c) {

//aquí va el cuerpo de la función

}

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.

Para darle un poco más de emoción, el programa no se limitará a escribir el resultado de la suma, sino que también escribirá todos los sumandos utilizados: 3 + 4 + 3 + 3 = 13.

La entradas es un número entero no negativo y menor que

public static void sumaDigitos(int numero) {

//cuerpo de la función

}

1. Un reloj digital de 24 horas, muestra las horas en el formato 00:00 HH:mm en un rango entre 00:00 y 23:59.

Realizar una función que reciba una hora en formato HH:mm (recibe dos datos, uno para la hora y otro para los minutos) y una duración (entero positivo) nos retorne la hora final, que es el producto de la suma de hora más los minutos.

Ejemplo: 10:30 + 140 = 12:40

public static void horasDuracion(int hh, int mm, int duracion) {

//cuerpo de la función

}

1. Construir una función que reciba dos horas en formato de 24 horas (similar al punto anterior) y nos muestre la suma de las horas.

Ejemplo: 9:15 + 6:50 = 16:05

17:20 + 10:05 = 1 dia y 03:25

public void sumaHoras (int hh1, int mm1, int hh2, int mm2) {

//cuerpo de la función

}

1. Construir una función que reciba una fecha (2 datos: mes y día) y nos la diga en forma de texto

Ejemplo: 02/23 y muestre “veintitrés de febrero”

public static void fechaEnTexto(int mes, int dia) {

// cuerpo de la función

}

1. Construir una función que reciba una fecha (mm/dd) y un número de días, y retorna la fecha final, producto de la fecha inicial más el número de días.

Ejemplo: fecha 03/26 , días 7 resultado 04/02

public static void sumaFechas(int mm1, int dd1, int mm2, int dd2) {

//cuerpo de la función

}

1. Construir una función que reciba 5 números enteros positivos y nos retorne su mediana, los números se pueden recibir en desorden.

Ejemplo: 21, 14, 6, 8, 20 Retorna 14

public static int mediana6Numeros(int a, int b, int c, int d, int e, int f) {

int mediana;

//cuerpo de la función

return mediana;

}

1. Construir una función que reciba tres números de un dígito y los combine mostrando todos los posibles números pares que se pueden construir con estos tres. Se construye sin ayuda de ciclos.

Ejemplo: 3 4 7 Salida 374, 734

2 3 6 Salida: 236, 362, 326, 632

public static void combinacionPares(int a, int b, int c) {

//cuerpo de la función

}

1. Realizar una función que calcula la suma descendente de un número dato de 5 cifras.

La suma descendente se realiza sumando los números producto de haber ido eliminando el dígito más significativo de forma reitera

Por ejemplo, la suma descendente del número 4.578 es igual a 4.578 + 578 + 78 + 8 = 5.242.

public static int sumaDescendente(int numero) {

//cuerpo de la función

return suma;

}

1. Llamamos números bicapicúa a aquellos que están compuestos por la concatenación de dos números capicúa. Recordemos que los números capicúa son aquellos que son los mismos si se leen de izquierda a derecha, como sí se leen de derecha a izquierda.

Algunos ejemplos de números bicapicúa son el 1.122, 1.211 o 121.343.

Construya una función que reciba un número entero positivo menor que 1.000.000.000 y diga si el número es bicapicúa o no.

public static boolean bicapicua(int numero)

boolean bandera = false;

//cuerpo de la función;

return bandera;

}

1. Construir una función que reciba un número entero positivo no superior a 1.000.000.000 e indique si el número es primo.

Por ejemplo: Dado el número 73 indicar, “Si es primo”

Dado el número 35 indicar, “No es primo”

public static void esPrimo(int numero){

//cuerpo de la función

}

CICLOS

1. Construya un algoritmo que genere la tabla de multiplicar de un número n.

2. Dado un ángulo en grados convertirlo a un ángulo menor de una vuelta

3. Construya un algoritmo que dado un número n, calcule el resultado de la serie:

1- ½+ 1/3 – 1/4+ 1/5 -….+1/n

4. Construya un algoritmo que dados dos números enteros, encuentre su cociente mediante restas sucesivas

5. Escriba un algoritmo calcule la potencia de ab.

potencial = a a a ... a b veces

6. Escriba un método que devuelva los primeros n tde la siguiente serie:

1, 8, 27, 64…

7. Escriba un algoritmo que devuelva los primeros n términos de la siguiente serie:

2 , 6, 8, 12, 14, 18…

8. Escriba un algoritmo que devuelva los primeros n términos de la siguiente serie:

-2 , 6, -8, 12, -14, 18…

9. Escriba un algoritmo que sume todos los n términos de la serie:

1, 1/2!, 1/3!, 1/4!...

10. Dado un número informar si es primo

11. Calcular el factorial de un número usando sumas sucesivas.

12. Haga un método para descomponer un número en sus factores primos.

13. Sumar los primeros n números primos. Hacer dos métodos (uno llamado isPrimo y otro llamado sumar)

14. Haga un programa que calcule la sumatoria los pares hasta n .

15. Haga un programa que diga si un número es o no perfecto. Un número N es perfecto si la suma de sus divisores (excluido el propio N) es N. por ejemplo 28 es perfecto, pues sus divisores (excluido el 28) son 1,2,4,7 y 14 su suma da 28.

16. Calcular el MCD de dos números

17. Hallar el mcm de dos números

18. Dado un programa que dado el precio del artículo y la cantidad de dinero entregado, indique las vueltas que deben darse, indicando el menor número de monedas. Usar restas sucesivas para el cálculo. Se sabe que se dispone de monedas de 100, 50, 20, 5 y 1.

19. Calcular la suma de 1 + 4 + 9 + 16 + 25 + ··· + n2

20. Imprimir el triángulo siguiente, pedir el n

n = 4

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

21. Construir un programa que visualice por pantalla todos los caracteres correspondientes a letras minúsculas.

22. Crear un programa que permita leer un número entero y lo invierta (Sin utilizar String) ej:

Entrada: 12345 Salida: 54321

23. Dado un número entero, retorna la suma de sus dígitos (Sin utilizar String) ej:

Entrada: 1234 Salida: 10

24. Se debe informar si dos números son números amigos. “Dos números se consideran amigos si son positivos y la suma de los divisores de uno es igual al otro. Se debe considerar que la suma de los divisores no debe incluir al número evaluado. Un ejemplo de números amigos son el 220 y el 284:

· Los divisores del 220 son: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 y 110, y suman 284;

· Los divisores del 284 son 1, 2, 4, 71 y 142, y suman 220[1]”.

**public** **static** **double** hallarSumaDivisores(**int** numero)

{**double** suma=0;

**return** suma;

}

**public** **static** **boolean** isAmigo(**int** numero1, **int** numero2)

{**boolean** centinela=**false**;

//Debe invocar hallarSumaDivisores

}

25. “Un número es un *cuadrado perfecto* si su raíz cuadrada es un número exacto (sin decimales). Por ejemplo, el 4 es un cuadrado perfecto (2²), al igual que lo son el 36 (6²) y el 3.500.641 (1871²).

Todos los números que *no* son cuadrados perfectos pueden multiplicarse por otros para conseguir serlo. Por ejemplo, el número 8 no es un cuadrado perfecto, pero al multiplicarlo por 2 se obtiene el 16, que sí lo es”[2].

La aplicación deberá recibir un caso de prueba y evaluar si el número dado corresponde a un cuadrado perfecto. Se debe el número más pequeño que al ser multiplicado por el número del caso de prueba da como resultado un cuadrado perfecto. Ejemplo:

Si se recibe como caso de prueba el 4 se debe devolver el 1 /\*4 Ya es cuadrado perfecto, se multiplica por 1 para no alterar su estado de cuadrado perfecto\*/

Si se recibe el 8 se devuelve el 2 /\* El 8 no es cuadrado perfecto, pero si se multiplica por 2 da 16 que si es cuadrado perfecto\*/

Si se recibe el 12 se devuelve el 3 /\* 12 no es cuadrado perfecto, pero si se multiplica por 3 se convierte en cuadrado perfecto\*/

**public** **static** **boolean** isCuadradoPerfecto(**int** num)

{//se saca raiz para verificar si es o no cuadrado.

//Haga casting para verificar que la raíz es un numero entero

}

/\*\*

\* Recibe un numero e informa el valor necesario para que sea convertido en cuadrado perfecto

\* **@param** num El numero a evaluar, num>=0

\* **@return** un numero entero

\*/

**public** **static int** completarCuadrado(**int** num)

{ /\*Aqui verificar si es cuadrado perfecto, si no hallar el valor requerido para que se convierta en uno\*/

}

[1] http://elmundodelosnumerosespeciales.blogspot.com/2016/04/numeros-amigos\_17.html

[2] Tomado de acepta el reto

| /\*\*  \* Permite leer un numero double mediante una caja de dialogo  \* **@param** mensaje El mensaje que verá el usuario  \* **@return** el numero ingresado por el usuario  \*/    **public** **static** **double** leerDouble (String mensaje)  {  **double** dato= Double.parseDouble(JOptionPane.showInputDialog(mensaje));  **return** dato;  }    /\*\*  \* Permite leer un numero entero mediante una caja de dialogo  \* **@param** mensaje El mensaje que verá el usuario  \* **@return** el numero ingresado por el usuario  \*/  **public** **static** **int** leerEntero (String mensaje)  {  **int** dato= Integer.parseInt(JOptionPane.showInputDialog(mensaje));  **return** dato;  }      /\*\*  \* Permite leer un caracter mediante una caja de dialogo  \* **@param** mensaje El mensaje que verá el usuario  \* **@return** el caracter ingresado por el usuario  \*/  **public** **static** **char** leerChar(String mensaje)  {  **char** dato= (JOptionPane.showInputDialog(mensaje)).charAt(0);  **return** dato;  }    /\*\*  \* Permite leer una cadena de caracteres mediante una caja de dialogo  \* **@param** mensaje El mensaje que verá el usuario  \* **@return** el string ingresado por el usuario  \*/  **public** **static** String leerString (String mensaje)  {  String dato= JOptionPane.showInputDialog(mensaje);  **return** dato;  }                /\*\*  \* Permite leer un boolean  \* **@param** mensaje El mensaje mostrado al usuario  \* **@return** un boolean, true si presiona el boton si, false en caso contrario  \*/  **public** **static** **boolean** leerBoolean (String mensaje)  {  **boolean** centinela=**false**;  **int** result = JOptionPane.showConfirmDialog(**null**,  mensaje,**null**, JOptionPane.**YES\_NO\_OPTION**);  **if**(result == JOptionPane.**YES\_OPTION**) {  centinela=**true**;  }  **return** centinela;  }    /\*\*  \* Permite imprimir un mensaje  \* **@param** mensaje El mensaje a imprimir  \*/  **public** **static** **void** imprimir (String mensaje)  {  JOptionPane.showMessageDialog(**null**,mensaje);  } |
| --- |

26. Implemente una solución que permita determinar si un código de barras en formato EAN-8 cumple con su especificación. Cada código EAN-8 contiene 8 dígitos y se gestiona como un String, esto teniendo en cuenta que con el código no se realizan operaciones aritméticas. Todo código de barras EAN-8, tiene asociado un código de comprobación (el último dígito). El dígito de comprobación para el código “12344567”, es 7. Para verificar si el código de barras en EAN-8 es correcto, se debe realizar lo siguiente:

- Se inicia por la derecha (se omite el código de control) y se suman los dígitos individuales, multiplicados por un factor (3 ó 1). Por ejemplo, para el código EAN-8: "65839522”, la operación a realizar es: 2 \* 3 + 5 \* 1 + 9 \* 3 + 3 \* 1 + 8 \* 3 + 5 \* 1 + 6 \* 3 = 88

- El código de barras es correcto, si se suma el dígito de control al resultado de la operación anterior. Esta sumatoria debe ser múltiplo de 10. Por ejemplo, la suma da 88+2=90. Si la suma es múltiplo de 10, entonces el dígito de control debe ser 0.

En la figura, se presenta la estructura de un código EAN-8. El cual fue generado desde el sitio web:<https://barcode.tec-it.com/es/EAN8>, y con fines únicamente académicos.



Figura. Representación código EAN-8

27. Se solicita escribir un algoritmo para informar si un código corresponde a un código de barras en formato E-11. Cada código de barras en formato E-11 contiene 11 dígitos. Todo código de barras tiene un código de comprobación (el primer dígito). El dígito de comprobación para el código “33456583722”, es 3. La estructura del código E-11 es 1-5-5. Un digito a la izquierda (“3”) y dos grupos de cinco dígitos (“34565”) y (“83722”). Por nomenclatura S1: (“34565”) y S2 (“83722”). Para verificar si el código de barras en E-11 es correcto, se debe realizar lo siguiente:

- Para cada grupo de cinco dígitos se inicia por la izquierda y se suman los dígitos individuales, multiplicados por un factor (5 o 3). El factor 5 se aplica para los dígitos pares y el factor 3 para los dígitos impares.

- El código de barras E-11 es correcto si cumple con las siguientes dos condiciones.

o El resultado de la suma de los dígitos de S1 es mayor que el resultado de la suma de los dígitos de S2.

o Se suma el digito de comprobación al resultado de la suma de S2. El resultado de la sumatoria debe ser un número primo (algoritmo recursivo).

Nota: para la implementación del método no se permite el uso de la instrucción: Integer.parseInt(parametro);

Tampoco se permite el uso de librerías de clase para gestión de números enteros o de punto flotante.

Determinar si un número es polidivisible. Los números *polidivisibles* son aquellos números que:

* Son mayores que cero.
* El número formado por su primer dígito es múltiplo de 1 (esto lo cumplen todos los números).
* El número formado por sus dos primeros dígitos es múltiplo de 2.
* El número formado por sus tres primeros dígitos es múltiplo de 3.
* El número formado por sus cuatro primeros dígitos es múltiplo de 4.
* Se repite sucesivamente para los dígitos y de acuerdo al tamaño de entrada.

En general, el número formado por sus *K* primeros dígitos es múltiplo de *K*. En la tabla, se presenta el análisis de dos números enteros que se desea determinar si son polidivisibles o no son polidivisibles.

Tabla. Números polidivisibles

| Número recibido | Análisis |
| --- | --- |
| 2016 | 2 es divisible por 1 (true) //substring de longitud 1  20 es divisible por 2 (true) //substring de longitud 2  201 es divisible por 3 (true) //substring de longitud 3  2016 es divisible por 4 (true) //substring de longitud 4  Como todas las dieron true y el número es mayor de cero, se concluye que es polidivisible |
| 4233 | 4 es divisible por 1 (true) //substring de longitud 1  42 es divisible por 2 (true) //substring de longitud 2  423 es divisible por 3 (true) //substring de longitud 3  4233 es divisible por 4 (false) //substring de longitud 4  En conclusión no es polidivisible |