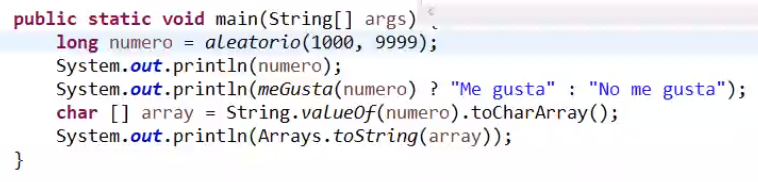
**EJERCICIO 1**

En una clase llamada Ejercicio1 escribe un programa que realice las acciones siguentes definiendo, además del método *main*, todos los métodos adicionales que se consideren necesarios:

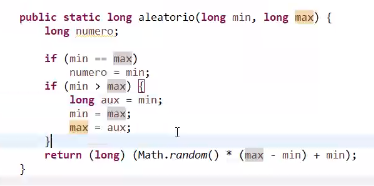
* Almacenar en una variable de tipo *long* un número aleatorio comprendido entre dos valores cualesquiera usando la clase *Math*.
* Sin usar sentencias *if* y sin convertir el número a cadena, determinar si el número obtenido me gusta. Un número sólo me gusta si:
  + La suma de los dígitos cuyo valor es impar es mayor que la suma del resto de dígitos.
  + No se cumple la condición anterior, pero el número es impar y l a suma de todos sus dígitos también lo es.
* Almacenar en un array de caracteres cada uno de los dígitos del número sin usar sentencias repetitivas.

Mostrar en la consola los resultados obtenidos en cada uno de los apartados anteriores.

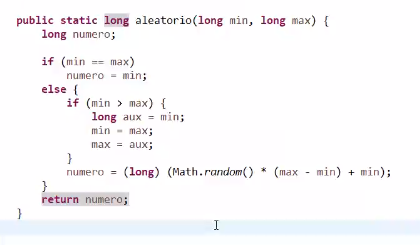
**METODO MAIN**



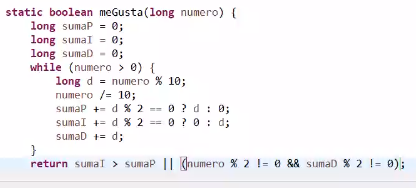
**CREAR UN NUM ALEATORIO**



**CREAR UN NUM ALEATORIO (ALTERNATIVA)**



**COMPROBAR SI ME GUSTA EL NUMERO**



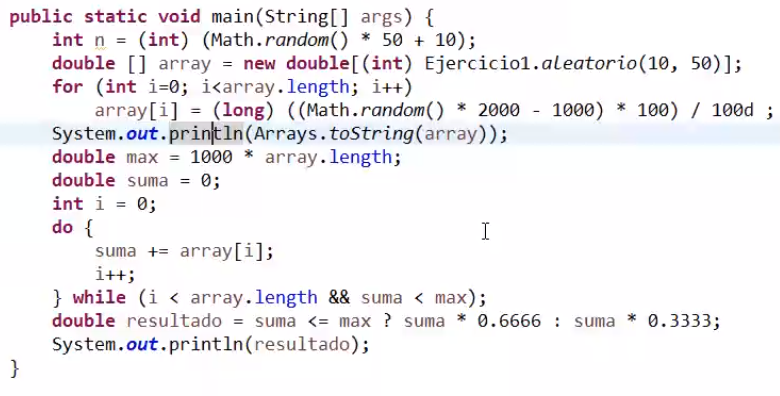
**EJERCICIO 2**

En una clase llamada Ejercicio2 escribe un programa que realice las acciones siguentes definiendo, además del método *main*, todos los métodos adicionales que se consideren necesarios:

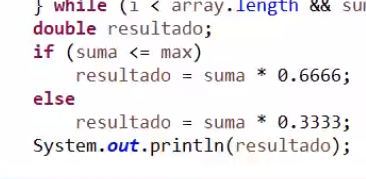
* Crear un array de un tamaño aleatorio entre 10 y 50.
* Usando la clase *Math*, almacenar en el array números aleatorios comprendidos entre –1000 y 1000 y con una precisión de 2 decimales.
* Obtener el resultado de procesar el contenido del array de la forma siguiente
  + Se sumarán los valores almacenados comenzando desde la posición cero en adelante.
  + Si en cualquier momento del proceso de cálculo de la suma ésta supera l amitad del valor máximo posible, el resultado final será el 33’33% del valor obtenido hasta ese momento, y no se sumarán el resto de los números almacenados en el array
  + Si no se produce la situación anterior, el resultado final será el 66’66% de la suma obtenida.

Mostrar en la consola el array y el resultado de procesar su contenido sin usar sentencias repetitivas.

**METODO MAIN**



((100d == 100’00))



((el final es equivalente a la imagen de arriba))

((no hace falta meter if dentro del do while porque la condición *suma < max* especifica que si se pasa de la mitad del maximo, va a parar el bucle))

((explicación del round))

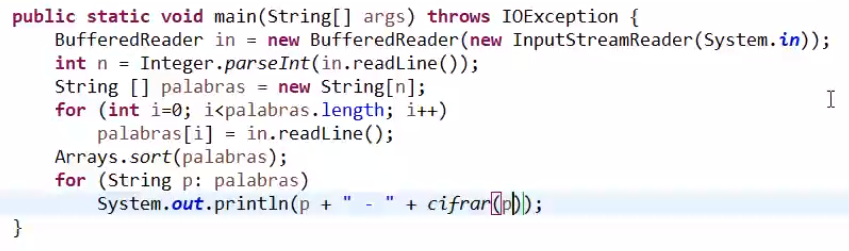


**EJERCICIO 3**

En una clase Ejercicio3 escribe un programa Java que lea de la consola una secuencia de cadenas y la procese según las especificacioens siguientes:

* Una vez leída la secuencia, se mostrará de nuevo en la consola imprimendo cada palabra en una línea y en orden alfabético.
* Cada palabra irá acompañada en la misma línea de un cifrado que se obtendrá con un método llamado *cifrar* de la forma siguiente:
  + Cada vocal se multiplica por la longitud de la cadena
  + Cada consonante se divide por la longitud de la cadena
  + Se suman todos los resultados obtenidos y esa será el resultado del cifrado
* Para llevar a cabo la lectura de los datos que el usuario escribe en la consola se asumen los supuestos siguientes
  + En la primera línea el usuario escribe el número *n* de palabras que escribirá a continuación
  + En la primera línea siempre escribe un número entero y nunca se equivoca
  + A continuación, escribe *n* palabras, cada una en una nueva línea
  + El usuario no está obligado a introducir las palabras en orden alfabético
  + No escribirá más de las palabras especificadas
* La solución del problema deberá estar basada en el uso de las clases *Integer, String y Arrays*.

**METODO MAIN**



**METODO CIFRAR**

