

No tiene sentido para cada reto ver su solución. Si fuera didáctico "ver soluciones " estarían publicadas. Está claro que la publicación masiva de las soluciones quitaría mucha enjundia al sentido de la página, no obstante, toda regla tiene una excepción y ahora mismo es interesante ver alguna solución para entender el funcionamiento del juez on line. También en nuestros boletines veremos soluciones adicionales para ilustrar algún concepto concreto pero sin olvidar que será un grupo reducido de soluciones con las que se va a trabajar.

1. ¡Hola mundo!

<https://www.aceptaelreto.com/problem/statement.php?id=116>

Este sencillo problema vale para: OBSERVAR QUE EL JUEZ SIEMPRE ENVÍA CASOS DE PRUEBA CORRECTOS. NO ES NECESARIO DEPURAR EL ENTRADA SALIDA CON IFs O CAPTURA DE EXCEPCIONES.

Hay que recordar que los casos de prueba que usa el juez siempre son correctos, por ello con esta frase

La entrada consta de una única línea que contiene un número n , $0 \leq n \leq 5$

deducimos que el juez nunca va enviar a nuestro programa una entero como 10, por lo tanto, en el siguiente código no es necesario el if, con sólo el for daría aceptada

```
import java.util.Scanner;

public class HolaMundo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        if (n >= 0 && n <= 5) {
            for (int i = 0; i < n; i++)
                System.out.println("Hola mundo.");
        }
    }
}
```

incluso ya que sabemos que n está siempre entre 0 y 5 podríamos resolverlo con un simple switch. Esto es una curiosidad, no es que sea importante ni interesante como técnica de programación pero constata el funcionamiento del juez

```
import java.util.Scanner;

public class HolaMundo {

    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int veces = teclado.nextInt();
        switch (veces) {
            case 5:
                System.out.println("Hola mundo.");
            case 4:
                System.out.println("Hola mundo.");
            case 3:
                System.out.println("Hola mundo.");
            case 2:
                System.out.println("Hola mundo.");
            case 1:
                System.out.println("Hola mundo.");
            case 0:
                System.out.println("Hola mundo.");
        }
    }
}
```

```

        System.out.println("Hola mundo.");
        case 2:
        System.out.println("Hola mundo.");
        case 1:
        System.out.println("Hola mundo.");

    }

}

```

Y por otro lado, la salida que genera mi programa se coteja con la salida que tiene el juez para sus casos de prueba por tanto si escribimos "Ola mundo." o "Hola mundo" (sin punto final) en el println() la salida no coincide.

2. ¿Qué lado de la calle_ id.217

<https://www.aceptaelreto.com/problem/statement.php?id=217>

Esta versión es aceptada.

Observa que el enunciado nos indica que las entradas. *"Se garantiza que nunca será mayor que 1.000"*. Esto nos vale para saber que podemos trabajar con int. En otros casos, necesitaríamos long o biginteger o trabajar directamente con Strings.

Recuerda que tal y como funciona el juez online para no tener que hacer depuración de la entrada/salida se garantiza que todos los casos de prueba que envía el juez son correctos, es decir, no va a enviar "hola" ni 2.5 ni 1001.

```

import java.util.Scanner;

public class QueLadoDeLaCalle {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int numero=sc.nextInt();
        while(numero!=0){
            if(numero%2==0){
                System.out.println("DERECHA");
            }else{
                System.out.println("IZQUIERDA");
            }
            numero=sc.nextInt();
        }
    }
}

```

Ahora decido hacer la lectura directamente en la condición en el bucle, pero me olvidé de eliminar la lectura del final del bucle que ahora no es necesaria. No hay errores de compilación pero hay un error lógico ya que en cada paso del bucle estoy haciendo dos lecturas.

```

import java.util.Scanner;
public class QueLadoDeLaCalle {
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int numero;
        while((numero=sc.nextInt())!=0){
            if(numero%2==0){
                System.out.println("DERECHA");
            }else{
                System.out.println("IZQUIERDA");
            }
            numero=sc.nextInt();//ii ahora sobra!!
        }
    }
}

```

Observo que funciona mal y el programa no termina ya que el 0 concuerda con la lectura del nextInt() de dentro del bucle.

```

L:\Programacion>java QueLadoDeLaCalle.java
3
10
41
IZQUIERDA
IZQUIERDA
0

```

PERO Si mando este código al juez me da RTE. Lo que típicamente provoca un RTE es una excepción java.

en cambio, si uso en mi terminal el método de sample.in si que se aprecia que salta excepción, debido a que se acabó el fichero y sigue un nextInt() a la espera

```

L:\Programacion>java QueLadoDeLaCalle.java <sample.in
IZQUIERDA
IZQUIERDA
Exception in thread "main" java.util.NoSuchElementException
    at java.base/java.util.Scanner.throwFor(Scanner.java:937)
    at java.base/java.util.Scanner.next(Scanner.java:1594)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2258)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2212)
    at QueLadoDeLaCalle.main(QueLadoDeLaCalle.java:6)

L:\Programacion>

```

CONCLUSIÓN: si el juez me da RTE, quiere decir que le saltó una excepción, si esta excepción no se aprecia metiendo los datos por teclado, podemos usar el método de sample.in para intentar ver la excepción generada