HORAS Y MINUTOS

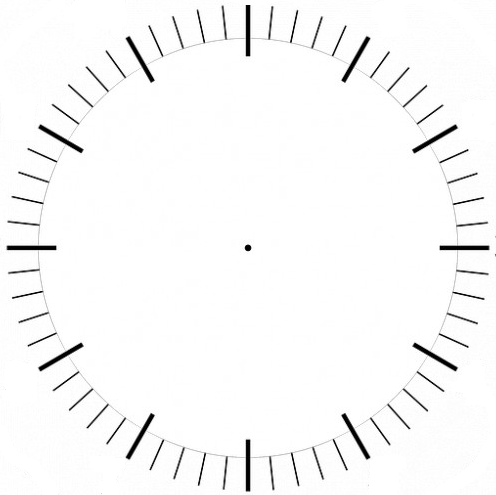
|  |  |
| --- | --- |
| Fecha de Elaboración: | Octubre 9, 2013 |
| Autores: | Diego Serrano |
| Fuente: | 12531 –Hours and Minutes  UVa Online Judge |

## PROBLEMA

Heidi tiene un reloj analógico discreto en la forma de un círculo, como el de la figura. Dos manecillas giran alrededor del centro del círculo, lo que indica horas y minutos. El reloj tiene 60 marcas colocadas alrededor de su perímetro, con distancia constante entre las marcas.

La manecilla de minuto se mueve de su marca actual a la siguiente exactamente una vez cada minuto. La manecilla de hora se mueve de su marca actual a la siguiente exactamente una vez cada 12 minutos, por lo que avanza cinco marcas cada hora.

Consideramos que ambas manecillas se mueven discretamente y al instante, lo que significa que siempre se sitúan exactamente sobre una de las marcas y nunca entre las marcas.



A la medianoche las dos manecillas al mismo tiempo llegan a la marca de arriba, lo que indica cero horas y cero minutos. Después de exactamente 12 horas o 720 minutos, las dos manecillas llegan a la misma posición de nuevo, y este proceso se repite una y otra vez. Tenga en cuenta que cuando la manecilla de minutos se mueve, la manecilla de las horas puede no moverse, sin embargo, cuando la manecilla de horas se mueve, la manecilla de minutos también se mueve.

A Heidi le gusta la geometría, y le gusta medir el ángulo mínimo entre las dos manecillas del reloj en diferentes momentos del día. Ella ha estado escribiendo algunas medidas, pero después de varios años y una lista larga, se dio cuenta de que algunos ángulos se repitieron, mientras que otros nunca aparecieron. Por ejemplo, la lista de Heidi indica que tanto a las tres, como a las nueve, el ángulo mínimo entre las dos manos es de 90 grados, mientras que un ángulo de 65 grados no aparece en la lista. Heidi decidió comprobar, para cualquier número entero entre 0 y 180, si existe al menos un tiempo del día de tal manera que el ángulo mínimo entre las dos manecillas del reloj es exactamente *A* grados. Ayúdala con un programa que responde a esta pregunta.

**Entrada**

Cada caso de prueba se describe en una línea. La línea contiene un entero *A* que representa el ángulo a comprobar (*0 ≤ A ≤ 180*).

**Salida**

Para cada caso de prueba, la salida es una línea que contiene un carácter. Si existe por lo menos una hora del día de tal manera que el ángulo mínimo entre las dos manecillas del reloj es exactamente *A* grados, se imprime la letra mayúscula 'Y'. Si no, imprime la letra mayúscula 'N'.

**Ejemplo de Entrada**

90

65

66

67

128

0

180

**Ejemplo de Salida**

Y

N

Y

N

N

Y

Y

## PLANTEAMIENTO DE LA SOLUCIÓN

Sabemos que nuestro reloj tiene 60 marcas igualmente separadas, lo cual quiere decir que para moverse una marca, la manecilla debe moverse 6 grados. Entonces, cada minuto, la manecilla de minutos se mueve 6 grados, y cada hora, la manecilla de horas se mueve 30 grados. Cada 60 minutos, la manecilla de minutos volverá a 0 grados. Y cada 12 horas, los ángulos se volverán a repetir.

Con los anteriores datos, podemos iterar desde las 0 horas 0 minutos, hasta llegar a las 12 horas, e ir marcando los ángulos que se han observado. Siendo así, calculamos el ángulo de la manecilla de las horas, multiplicando el número de horas por 30. Para el ángulo de la manecilla de los minutos, multiplicamos el número de minutos por 6. Con estos datos, calcular el ángulo mínimo entre las manecillas es trivial.

Parte importante de este ejercicio, es que para optimizar el tiempo de respuesta se pre-calcularán cuales ángulos pueden formarse en un reloj análogo. De esta manera, el tiempo para responder cada caso de prueba es constante *O(1)*.

La solución fue enviada a Uva Online Judge el 8 de octubre de 2013, con un tiempo de ejecución de 0.162 segundos.

## CODIGO FUENTE

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.util.Arrays;

public class Main {

public static boolean[] angles = new boolean[360];

public static void main(String[] args) throws IOException {

BufferedReader reader;

Arrays.fill(angles, false);

int h = 0;

int m = 0;

while (h < 12) {

int angleH = 30 \* h;

int angleM = 6 \* m;

int a = Math.abs(angleH - angleM);

if (a > 180) {

a = a - 180;

}

angles[a] = true;

m++;

if (m % 60 == 0) {

m = 0;

h++;

}

}

InputStream input = System.in;

reader = new BufferedReader(new InputStreamReader(input));

String params;

while ((params = reader.readLine()) != null) {

int a = Integer.parseInt(params);

if (angles[a]) {

System.out.println("Y");

} else {

System.out.println("N");

}

}

reader.close();

}

}