



NORMAS DE ENTREGA

LEA DETENIDAMENTE LAS NORMAS DE LA PRUEBA

Cualquier incumplimiento de las normas significará una respuesta nula en el examen y por tanto una valoración de 0 puntos.

- La resolución de los ejercicios propuestos se entregará en un único archivo formato ZIP que contendrá **EXCLUSIVAMENTE** los proyectos de Netbeans que compongan la solución a los problemas planteados.
- El nombre del fichero tendrá un formato específico dictado por el nombre de cada alumno. Por ejemplo, para un alumno llamado “José María Núñez Pérez” el fichero se nombrará como NunyezPerezJM.zip. Obsérvese que las tildes son ignoradas y las eñes sustituidas.
- El fichero se subirá utilizando la correspondiente tarea en el aula virtual.

IMPORTANTE: Cualquier envío que no respete el formato de compresión o el nombre adecuado será ignorado y, por tanto, valorado con cero puntos.



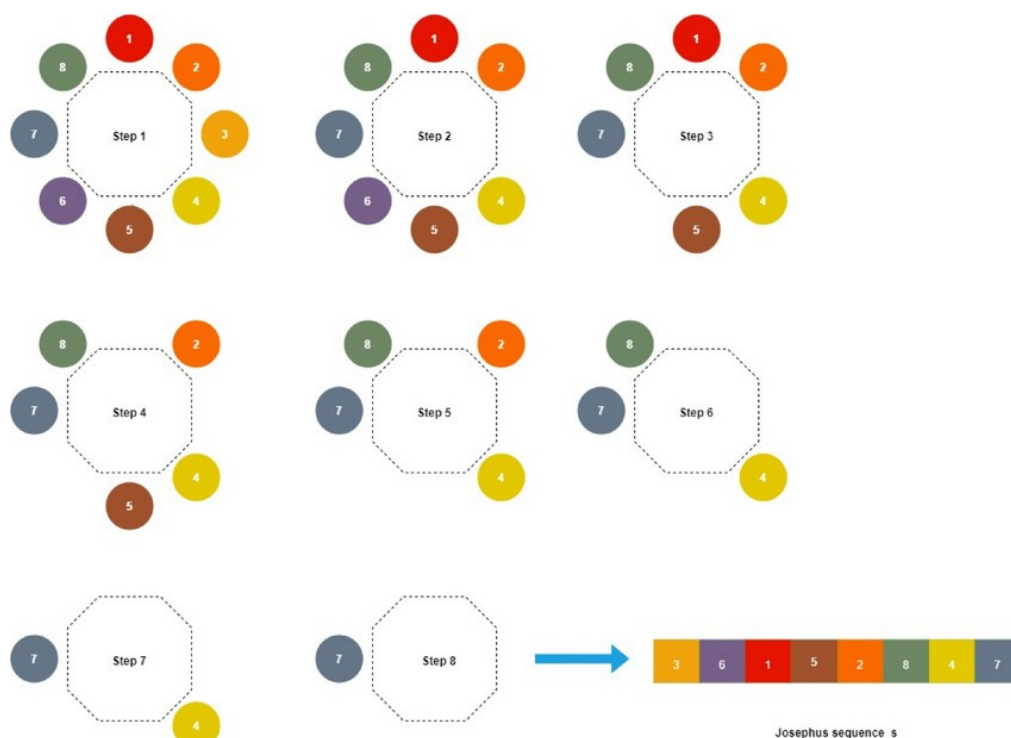
Problema 1

En el siglo I d.C. se produjo una revuelta judía contra Roma. La lucha se dispersó en muchos frentes, en uno de ellos el historiador judío Josephus y 39 de sus compañeros se enfrentaron a los romanos refugiándose en una cueva. Como la derrota era inminente, decidieron que preferían morir antes que ser esclavos de los romanos. Por ello decidieron suicidarse de forma sistemática. Se organizarían en un círculo, designando a uno de ellos como número uno y procedieron en sentido horario matando a cada séptimo hombre. Josephus fue entre otras cosas un consumado matemático, por lo que instantáneamente descubrió dónde debería colocarse para ser el último en morir. Pero cuando llegó el momento, en lugar de suicidarse, se unió al bando romano.

El objetivo del problema es implementar un programa que averigüe en qué posición se debe colocar Josephus para no ser asesinado. La solución debe generalizarse para cualquier cantidad de soldados judíos y cualquier paso, entendiendo por *paso*, el número k de soldados judíos que hay que saltar en el círculo hasta seleccionar al siguiente en morir. En el problema real el número de soldados fue cuarenta ($n = 40$) y el paso fue siete ($k = 7$).

El programa deberá preguntar por teclado el número de soldados judíos (n), así como el paso (k), debiendo devolver la posición exacta en la que se debería colocar Josephus para ser el último en morir y así sobrevivir, mostrando durante la ejecución mensajes informativos con el número del soldado judío que se ejecuta en cada iteración.

A continuación se muestra un ejemplo gráfico de cómo sería este problema para ocho soldados ($n = 8$) y paso dos ($k = 2$)



La salida de la ejecución del programa para este ejemplo sería:

```
Número de soldados (n): 8
Paso (k): 2
Cargados los 8 soldados en la cola
Se ejecuta al soldado 3
Se ejecuta al soldado 6
Se ejecuta al soldado 1
Se ejecuta al soldado 5
Se ejecuta al soldado 2
Se ejecuta al soldado 8
Se ejecuta al soldado 4
Sobrevive el soldado 7
```



Requisitos:

1. Desarrollar la clase *ArrayQueue* que implemente la interfaz *IQueue*. La cola tendrá que ser implementada usando un array de tamaño 1000.

```
public interface IQueue<E> {  
    boolean find(E o);  
    E dequeue() throws EmptyQueueException;  
    void enqueue(E o);  
    E front() throws EmptyQueueException;  
    boolean isEmpty();  
    int size();  
}
```

2. Desarrollar la clase *JosephusGame* que, haciendo uso de objetos de la clase *ArrayQueue* implementada en el apartado anterior, resuelva el problema de Josephus tal y como se ha descrito en este enunciado.