

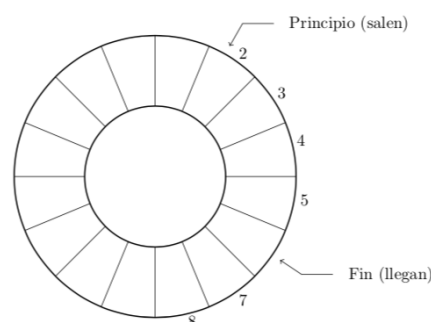
Implementación con array circular de una cola

El TAD **queue** implementa una estructura de datos lineal cuya principal característica es que el acceso a los elementos se realiza en el mismo orden en que fueron almacenados. El TAD ofrece 4 operaciones además de su constructor: **push** permite guardar un dato en la cola, **pop** permite eliminar un elemento de la cola si esta no está vacía, **front** nos devuelve el primer elemento de la cola si esta no está vacía y **empty** nos indica si la cola está vacía.

El TAD **queue** se puede implementar por medio de una lista enlazada simple con punteros al comienzo y final de la lista o con un array dinámico. En ambos casos se puede obtener un coste constante para todas las operaciones.

En la implementación con array dinámico se debe utilizar un array circular, es decir un array en que el siguiente al último elemento sea el primero. El acceso al siguiente elemento se implementa utilizando aritmética modular. El elemento siguiente a la posición i es la posición $(i+1)\%c$ siendo c la capacidad a la que está declarado el array.

En este problema vamos a desarrollar una implementación del TAD **queue** por medio de un array circular y lo probaremos con un programa que creará una cola con una serie de valores de la entrada. Dará una vuelta completa a la cola, duplicará los elementos de la cola y por último irá sacando los elementos mientras los muestra.



Requisitos de implementación.

Recuerda que la implementación que se proporciona en el campus para el TAD **stack** utiliza un array dinámico como tipo representante y la implementación del TAD **queue** utiliza como tipo representante una lista enlazada simple con punteros al principio y al final de la lista.

El programa para probar el tipo implementado se proporciona a continuación del sample. Se puede descargar desde el juez en la pestaña **ProblemaSet**, pulsando **Input**.

Entrada

La entrada está formada por una serie de casos de prueba, cada uno en una línea. Cada línea contiene los caracteres que forman ese caso. La entrada de datos termina con una línea que contiene solo el carácter punto y que no debe procesarse.

Salida

Para cada caso de prueba se escribirá en una línea la cadena de entrada después de haber ejecutado el programa que se proporciona a continuación del sample.

Entrada de ejemplo

```
Hola Mundo!!
Esta linea termina en un punto.
Los casos terminan con un solo .

La linea anterior estaba vacia
.
```

Salida de ejemplo

```
Hola Mundo!!  
Esta linea termina en un punto.  
Los casos terminan con un solo .  
  
La linea anterior estaba vacia
```

Autor: Isabel Pita