Ejercicios Tipos Lineales.

1. Implementar una clase Pila usando una lista enlazada.

Esta implementación de pila utiliza una clase privada anidada Nodo como base para representar la pila como una lista enlazada de objetos Nodo.

La variable de instancia first se refiere al primer nodo (el más recientemente insertado) de la lista enlazada. La variable de instancia next en cada Nodo se refiere al Nodosucesor (el valor de next en el nodo final es null). No se necesitan constructores explícitos, porque Java inicializa las variables de instancia a null.

Implementar los métodos: empty, peek, push, pop y search.

Method Summary

All Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
boolean	empty()	Tests if this stack is empty.
E	peek()	Looks at the object at the top of this stack without removing it from the stack.
E	pop()	Removes the object at the top of this stack and returns that object as the value of this function.
E	push(E item)	Pushes an item onto the top of this stack.
int	search(Object o)	Returns the 1-based position where an object is on this stack.

2. Implementar un Deque.

Una cola de doble extremo o deque (pronunciado "deck") es una colección que es una combinación de una pila y una cola.

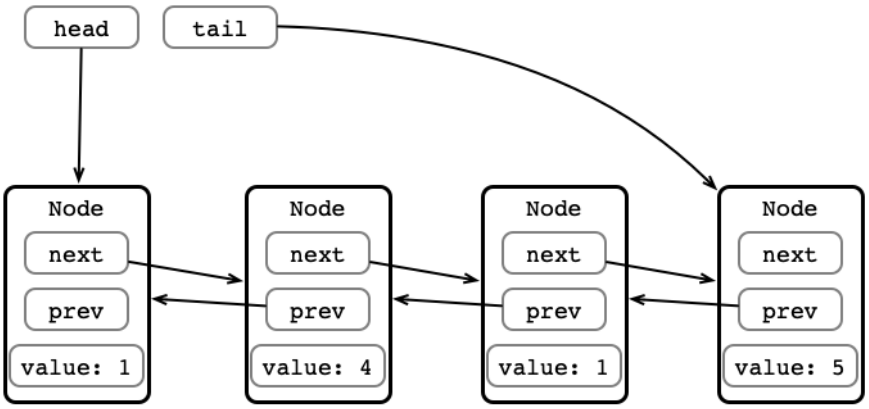
Escribe una clase Deque que utilice una lista enlazada para implementar la siguiente API:

- public class Deque<Item>
- Deque() crea un deque vacío
- boolean isEmpty() ¿está vacío el deque?
- void enqueue(Item item) añade item al final
- void push(Item item) añade el elemento al principio
- Item pop() elimina y devuelve el elemento al principio
- Item dequeue() elimina y devuelve el elemento al final

https://docs.oracle.com/en/java/javase/19/docs/api/java.base/java/util/Deque.html

Summary of Deque methods

	First Element (Head)		Last Element (Tail)	
	Throws exception	*Special value*	*Throws exception*	*Special value*
Insert	`addFirst(e)`	`offerFirst(e)`	`addLast(e)`	`offerLast(e)`
Remove	`removeFirst()`	`pollFirst()`	`removeLast()`	`pollLast()`
Examine	`getFirst()`	`peekFirst()`	`getLast()`	`peekLast()`



3. El problema de Josephus

Cuenta la leyenda: n personas están en una situación desesperada y acuerdan la siguiente estrategia para reducir la población. En se disponen en círculo (en posiciones numeradas de 0 a n1) y proceden a alrededor del círculo, eliminando a cada m personas hasta que sólo quede una.

La leyenda cuenta que Josephus descubrió dónde sentarse para evitar ser eliminado.

Escribe un cliente Cola Josephus que tome dos argumentos enteros m y n e imprime el orden en el que las personas son eliminadas (y así mostraría a Josephus dónde sentarse en el círculo).

java Josephus 2 7

1 3 5 0 4 2 6

Josephus 3 12

2 5 8 11 3 7 0 6 1 10 4 9