**Control de lectura 03-AYED**

Jefer Alexis González Romero. LCAT-2. 12/10/2021

**1.** Considere una pila S y una cola Q, ambas vacías en un principio. Ilustre el paso a paso de las siguientes operaciones.

* S.push(2) → S.push(8) → S.push(11) → S.pop() → S.push(3) → S.push(7) → S.pop() → S.pop()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

S

S.push(2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 |  |  |  |  |

S

S.push(8)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 8 |  |  |  |

S

S.push(11)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 8 | 11 |  |  |

S

S.pop()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 8 | 11 |  |  |

S

S.push(3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 8 | 3 |  |  |

S

S.push(7)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 8 | 3 | 7 |  |

S

S.pop()

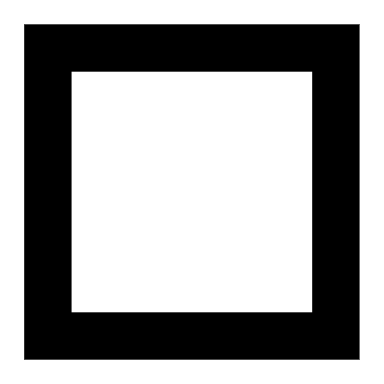
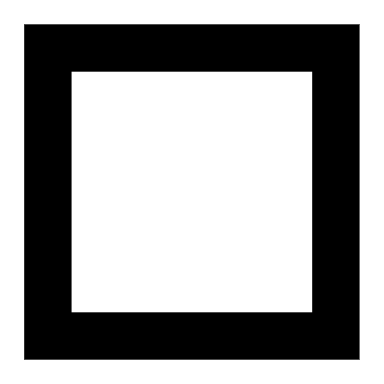
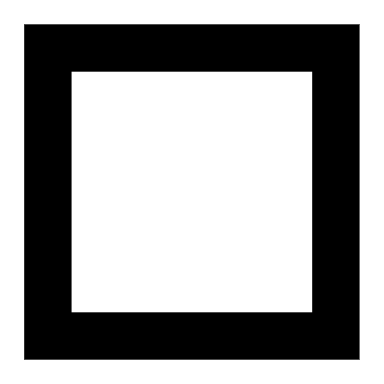
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 8 | 3 | 7 |  |

S

S.pop()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 8 | 3 |  |  |

S

* Q.enqueue(4) → Q.enqueue(17) → Q.enqueue(20) → Q.enqueue(6) → Q.dequeue() → → Q.dequeue() → Q.enqueue(-5)→ Q.dequeue()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |

Q

Q.enqueue(4)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 4 |  |  |

Q

Q.enqueue(17)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 4 | 17 |  |

Q

Q.enqueue(20)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 4 | 17 | 20 |

Q

Q.enqueue(6)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 |  | 4 | 17 | 20 |

Q

Q.dequeue()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 |  | 4 | 17 | 20 |

Q

Q.dequeue()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 |  |  | 17 | 20 |

Q

Q.enqueue(-5)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | -5 |  |  | 20 |

Q

Q.dequeue()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 6 | -5 |  |  | 20 |

Q

**2.** Desarrolle un algoritmo que identifique si una cadena de texto contiene una lista de paréntesis correctamente anidados y balanceado.

Adjunto código con nombre “2”.

**3.** Diseñe una función para invertir la dirección de una lista enlazada simple, es decir, una función que invierta todos los punteros entre los elementos de la lista. El algoritmo debe tener complejidad O(*n*).

Adjunto código con nombre “3”. Donde la función se llama “invert”.

**4.** Modifique el código de la lista enlazada para que sea una doble lista enlazada, e implemente las siguientes funciones:

Adjunto código con nombre “4”.

* Insertar un nuevo elemento

La función en el código se llama “insert”

* Eliminar un elemento dado su valor (considere el caso de borrar la cabeza de la lista)

La función en el código se llama “delete\_element”

* Eliminar los elementos duplicados

La función en el código se llama “delete\_duplicate”

* Unir dos listas

La función en el código se llama “unir”