QUESTION 1
For a method in ArrayQueue, replace the highlighted parts by the correct one for the method exist. The method will check if the element elem exist in the queue. Returns true if it exist, false otherwise, method should start search from head.
The name of array is data public briotlean exist (T. e.lem). temp = head; for(int i = 0; 1 , 3; 30; 1++) return true; [nodes[Temp]

temp = (Temp + 1) Maxsi30) return false;

QUESTION 2

1. The Big-Oh for the methods push, pop, empty, and full of the Stack for both implementations (Linked-List/Array) is:

All O(1)

None

push/pop O(N), empty/full O(1)

All O(n)

QUESTION 3

1. The behavior of adding and removing elements in Queue is:



LIFO

FIFO

As a user of the Stack ADT, consider the static method print, that takes a stack's containing data of type String, and prints its elements from top to bottom. Stack's should be unchanged after the method is done Complete the code below by choosing the correct answer:

```
    public static void print(Stack<String> s) (

       System.out.println(e);
       10.
```

Line 2:

- Stack <String> s2 = new LinkedStack<String>();
- Queue < String > q = new LinkedQueue < String > ();
- List <String> | = new List <String>();
- Node<String> tmp = s.top;
- None

- The behavior of adding and removing elements in stacks is:

 - FIFO
- C FIFA
- C LIFO
- 1. As a user of the Stack ADT, consider the static method print, that takes a stack's containing data of type String, and prints its elements from top to bottom. Stack s should be unchanged after the method is done. Complete the code below by choosing the correct answer:
 - 1. public static void print(Stack<String> s) { 3. 4.

 - System.out.println(e);
 - 7.
 - 8.
 - . . .
 - 10.}

Line 3:

- while(!s.last())
- while(tmp != null)
- while(!s.empty())
- None
- for(int i = 0; i < s.length(); i++)

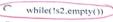
QUESTION 6

1. The result of evaluating the postfix expression: 2 3 4 + 1 * -

As a neer of the Stack AET, consider the static method print, that takes a stack's containing data of type String, and prints its elements from top to bottom. Stack's should be unchanged after the method is done. Complete the code below by choosing the correct answer:

Line 8:

for(int i = 0; i < q.length(); i++)



None

c tmp = top;

while(!l.last())

QUESTION 10

1. The expected behavior of this method is to:

```
public static void method2(Stack s) {
    Stack s1 = new LinkedStack ();
    int n = 0;
    while(!s.empty()) {
        s1.push(s.pop());
        n++;
    }
    n = n / 2;
    for(int i = 0; i < n; i++)
        s.push(s1.pop());</pre>
```

- Not change anything in stack s
- C Keep the top half the stack s
- Keep the bottom half of the stack s
- Reverse the stack s

QUESTION 8

As a user of the Stack ADT, consider the static method print, that takes a stack's containing data of type.

String, and prints its elements from top to bottom. Stack's should be unchanged after the method is done. Complete the code below by choosing the correct answer:

Line 9:

- s.push(l.retrieve());
- top.next = tmp;
- s.push(q.serve());
 - s.push(s2.pop());
- None

```
10.)
    Line 4:
       T e = s.pop();
       T e = s.getData();
        Te = s.retrieve();
    Te = tmp.data;
        None
   QUESTION 13
1. The expected behavior of this method is to:
   public static void method1(Stack s) {
       Stack s1 = new LinkedStack ();
       Stack s2 = new Linked Stack ();
        while(!s.empty())
            sl.push(s.pop());
        while(!sl.empty())
            s2.push(s1.pop());
        while(!s2.empty())
             s.push(s2.pop());
        Reverse the stack s
         Keep the bottom half of the stack s
         Keep the top half of the stack s
         Not change anything in stack s
```

```
10.)
    Te=s.pop();
     Te = s.getData();
    Te = s.retrieve();
    O Te = tmp.data;
    C None
   QUESTION 13
1. The expected behavior of this method is to:
  public static void method1(Stack s) {
      Stack sl = new LinkedStack ();
      Stack s2 = new Linked Stack ();
      while(!s.empty())
          sl.push(s.pop());
      while(!sl.empty())
         s2.push(s1.pop());
     while(!s2.empty())
          s.push(s2.pop());
     Reverse the stack s
     Keep the bottom half of the stack s
     Keep the top half of the stack s
     Not change anything in stack s
```

QUESTION 17 As a user of the Stack ADT, consider the static method print, that takes a stack's containing data of type string, and prints its elements from top to bottom. Stack's should be unchanged after the method is done, complete the code below by choosing the correct answer: public static void print(Stack<String> s) (2. ... { . . . System.out.println(e); 10.} Line 5: None tmp = tmp.next;l.insert(e); q.enqueue(e); s2.push(e); 1. Suppose we have a circular array implementation of the queue class, with 8 elements in the queue stored at data[4] (least recent element) through data[11] (most recent element). The Maximum Size is 15. Where does the enqueue member function place the new element in the array? data[14]z data[0] data[11] c data[12]

OUESTION 7

What does his method de?

Static <T> void method2 (Queue<T> q, T elem) {

int i = 0;

int i = 0;

int i = 0;

while (i < n) {

Yellem.equals (temp);

if (!elem.equals (temp))

if (!elem.equals (temp);

q.enqueue (temp);

i++;

}

Infinite loop

Remove all elements equal to elem

Won't work correctly, when an element is removed, we need to add i-
Won't work correctly, need to remove !(not symbol) in the if statement

QUESTION 14

1. In the circular array version of the queue class (with a fixed-sized array), which operations require O(n) time for their worst-case behavior?

c full

Serve

enqueue

c None

QUESTION 15

1. A data structure in which elements can be inserted or deleted at/from both the ends but not in the middle is

Priority queue

Queue

C Dequeue

C Circular queue

1. The Big-Oh method enqueue an element in linkedimplementation of priority queue is?

○ O(nlogn)

O(logn)

C O(n)

 $_{\circ}$ O(n2