NAME: AZUBIKE SAMSON AVWEROSUOGHENE
MATRIC NO: 2023/253812
DEPARTMENT: COMPUTER SCIENCE

```
// Source Code
class CircularQueue {
    private char[] queue;
    private int front, rear, size;

    public CircularQueue(int size) {
        this.queue = new char[size];
        this.front = -1;
        this.rear = -1;
        this.size = size;
    }

    // Check if the queue is full
    public boolean isFull() {
        return (rear + 1) % size == front;
    }

    // Check if the queue is empty
    public boolean isEmpty() {
        return front == -1;
    }

    // Enqueue an element
    public boolean enqueue(char element) {
        if (isFull()) {
            System.out.println("Queue is full");
            return false;
        }
        if (isEmpty()) {
            front = 0;
        }
        rear = (rear + 1) % size;
        queue[rear] = element;
        return true;
    }

    // Dequeue an element
    public int dequeue() {
        if (isEmpty()) {
            System.out.println("Queue is empty");
            return -1;
        }
        char removedElement = queue[front];
        if (front == rear) {
            // Reset the queue when it becomes empty
            front = -1;
            rear = -1;
        } else {
            front = (front + 1) % size;
        }
        return removedElement;
    }

    public int getFirst() {
        if (isEmpty())
```

```
            return -1;
        return queue[front];
    }

    public int getLast() {
        if (isEmpty())
            return -1;
        return queue[rear];
    }

    // Display the queue
    public void display() {
        if (isEmpty()) {
            System.out.println("Queue is empty");
            return;
        }
        System.out.print("Queue elements: ");
        int i = front;
        while (true) {
            System.out.print(queue[i] + " ");
            if (i == rear)
                break;
            i = (i + 1) % size;
        }
        System.out.println();
    }

}

class CircularQueueImpl {

    public static void main(String[] args) {
        CircularQueue circularQueue = new CircularQueue(10);

        circularQueue.enqueue('A');
        circularQueue.enqueue('B');
        circularQueue.enqueue('C');

        System.out.println("\n\n");

        while (!circularQueue.isFull()) {
            int rand = (int) (Math.random() * 10);

            if (rand > 1) {
                int nextLetter = circularQueue.getLast() + 1;
                circularQueue.enqueue((char) nextLetter);
            } else {
                circularQueue.dequeue();
            }
        }

        circularQueue.display();
    }
}
```

// Output

```
Samie Azubike@SAMIE-AZUBIKE MINGW64 ~/IdeaProjects/learning
$  cd c:\\Users\\Samie\ Azubike\\IdeaProjects\\learning ; /usr/bin/env C:\\Program\ Files\\Java\\jdk-21\\bin\\java.exe
 -XX:+ShowCodeDetailsInExceptionMessages -cp C:\\Users\\Samie\ Azubike\\AppData\\Roaming\\Code\\User\\workspaceStorage
\\4a09fd244e2bcae82ca3af645ebf1cb4\\redhat.java\\jdt_ws\\learning_9ceeaaa6\\bin CircularQueueImpl
ser\\\\workspaceStorage\\\\4a09fd244e2bcae82ca3af645ebf1cb4\\\\redhat.java\\\\jdt_ws\\\\learning_9ceeaaa6\\\\bin Circu
larQueueImpl ;9a2057a8-5b88-4484-920b-80d5fe42d103

Queue elements: B C D E F G H I J K
```