# UNIVERSITY INSTITUTE OF COMPUTING

### PROJECT REPORT

\*\*ON\*\*

\*\*QUEUE-BASED EDUCATIONAL QUIZ SYSTEM\*\*

\*\*Program Name:\*\* BCA

\*\*Subject Name/Code:\*\* Data Structures (23CAT-201)

---

\*\*Submitted by:\*\*

\*\*Name:\*\* Siddharth

\*\*UID:\*\* 23BCA10454

\*\*Section:\*\* BCA – 3 “A”

\*\*Group:\*\* 5

\*\*Submitted to:\*\*

\*\*Name:\*\* Mrs. Shilpi Mittal

\*\*Designation:\*\* Co-ordinator

---

**ABSTRACT**

### Introduction:

This C-based quiz application is designed to help students practice and understand the **Queue data structure** through a gamified learning system. It demonstrates both **educational content delivery** and **data structure implementation**, specifically focusing on **queue applications**. The program tests the user's understanding with 10 multiple-choice questions, explanations for each, and a grading system based on performance.

### Technique:

This project was implemented in **C language**, using:

1. **Structures (`struct`)** to define a `QuizQuestion` object with attributes for the question, options, correct answer, and explanation.
2. **Control Structures & Loops** for quiz flow, answer validation, and tracking remaining lives.
3. **Queue Concept Integration** as the educational theme of the quiz—questions test real-world applications of different queue types (Simple Queue, Circular Queue, Deque, Priority Queue).
4. **Grading Logic** to evaluate performance and provide motivational feedback based on score percentages.

### System Configuration:

* **OS:** Android (Termux) / Linux (online compiler)
* **Compiler**: GCC
* **Memory**: Lightweight, under 1 MB usage
* **Editor**: GitHub Mobile + Online C Compilers

**SUMMARY**

### Input:

The user is guided through the following:

* Instructions and welcome screen.
* Answering each of the **10 questions**, with options (a-d).
* A total of **5 lives** are given. A wrong answer deducts a life.
* After each question, an **explanation** is displayed to reinforce learning.

### Sample Input:

```

Question 1/10: Which of the following queues uses FIFO?

a) Circular Queue

b) Simple Queue

c) Priority Queue

d) Double-Ended Queue

Your answer (a-d): b

✅ Correct!

Explanation: A simple queue follows the FIFO rule...

```

---

**Process**

### *Algorithm:*

1. **Initialize** quiz array with 10 `QuizQuestion` structs containing:

* A Question
* 4 Answer Options
* The Correct Answer
* The Explanation

1. **Set** initial values:

* Score = 0
* Lives = 5

1. **For** each question in the quiz:

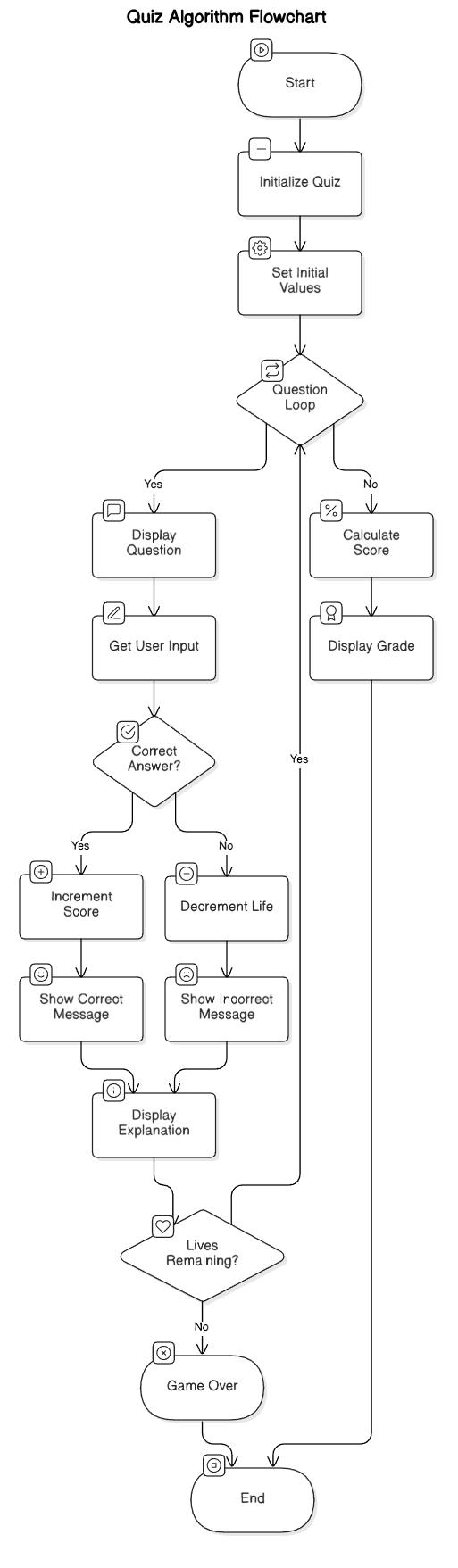
* Display the question and its options.
* Get user input.
* **If** the answer is correct
  + Increment score
  + Show "Correct!" message
* **Else:**
  + Decrement a life.
  + Show "Incorrect!" message and remaining lives.
* Display the explanation.

1. **If** lives = 0:

* End quiz early and show "Game Over".

1. **After all questions or lives end:**

* Calculate score percentage.
* Display grade and motivational quote.



*Figure 2: Algorithm Flowchart*

**OUTPUT**

- \*\*Correct/Incorrect response\*\* shown immediately.

- \*\*Lives tracking\*\* on wrong answers.

- \*\*Explanations\*\* after each question.

- \*\*Final Score and Grade\*\* with feedback.

Example:

```

Final Score: 8/10 (80%)

Grade: B-

Quote: "Good work! Just a little more practice needed!"

```

**UNIQUE ASPECTS**

- \*\*Educational Use of Queues:\*\* The project itself teaches queue concepts, going beyond just using data structures.

- \*\*Feedback-Oriented Learning:\*\* Each wrong answer still provides the correct explanation.

- \*\*Gamified Engagement:\*\* Using lives and grades to simulate a "game-over" system keeps users engaged.

- \*\*Expandable Template:\*\* The code structure allows easy addition of more topics (e.g., Stacks, Trees, Sorting).

**FUTURE SCOPE**

- Convert this CLI quiz into a \*\*web-based app\*\* using AJAX or API methods.

- Allow \*\*custom quiz topics\*\*, \*\*question banks\*\*, and \*\*user login\*\* to track progress.

- Integrate with \*\*GitHub Pages\*\* for online access to educational tools on Data Structures.