Golang Programming Fundamentals

Assignment #1 Due Date:

Convert C Source Code to golang

Programming Requirements

- Submit your assignment into a **single golang source file**.
- You must write **your name** at the top of your assignment source list.
- Write your **compiler version and operating system name** at the top of your assignment source list.
- Assignment that is turned in late will lose **one point per day** starting after the **due date**.
- Make sure that you do appropriate **error checking** in your program. (User-friendliness)
- Do not turn in **incomplete or crashing program**, you will receive **zero points**.
- Do not **import third part packages** into your golang source file your assignment will not be graded.
- Make sure to read **grading policy** carefully that will tell you how your assignment is graded.

Assignment #1 Grading Policy

Category	**Points Possible**	**Points Received**
Correctness and Efficiency	10	
Meaningful variable names	10	
Use **Array** to implement Queue	15	
Use **iota** to identify constants that are used in switch statement.	20	
Style and code readability	20	
Complete Documentation	10	
User-friendliness, see example 2-3	15	
Total	100	

Assignment Description

In assignment #1 you will learn how to **convert C** source code into a **golang** code. You will implement a Queue using an array of integer. Use three functions for three operations like insert, delete, and display. It is your responsibility to write correct logic for three operations in golang and organize the source code the best possible way, **without changing** the **logic** of the program. Please follow the **grading policy** while you are doing assignment one.

Program Explanation

1. Ask the user for the operation like insert, delete, display and exit.
    1. **Insert element to queue**
    2. **Delete element from queue**
    3. **Display all elements of queue**
    4. **Quit**

2. According to the option entered, access its respective function using switch statement. Use the variables front and rear to represent the first and last element of the queue.

3. In the function insert(), firstly check if the queue is full. If it is, then print the output as "Queue Overflow". Otherwise take the number to be inserted as input and store it in the variable add_item. Copy the variable **add_item** to the array **queue_array[]** and increment the variable rear by 1.

4. In the function delete(), firstly check if the queue is empty. If it is, then print the output as "Queue Underflow". Otherwise print the first element of the array queue_array[] and decrement the variable front by 1.

5. In the function display(), using for loop print all the elements of the array starting from front to rear.

6. Exit.
Quit your program with happy message.