**📘 README: Custom Stack and Queue Implementation**

**🛠 Language and Version**

This project is implemented in **C# (C Sharp)** using **.NET 6.0**.

**🚀 How to Run the Program**

1. **Prerequisites**:
   * [.NET SDK 6.0+](https://dotnet.microsoft.com/en-us/download)
   * A C# IDE or editor (e.g., Visual Studio, Visual Studio Code)
2. **Steps**:
   * Clone or download the project files.
   * Open the project folder in your IDE.
   * Build the solution to restore dependencies and compile the code.
   * Run the program using the IDE's run command or via terminal:
   * dotnet run
3. **Usage**:
   * The program demonstrates both Stack and Queue operations.
   * You can modify the Main method to test different scenarios.

**📦 Project Structure**

* CustomStack.cs: Implements a generic Stack class using a raw array.
* CustomQueue.cs: Implements a generic Queue class using a raw array.
* Program.cs: Contains sample usage and test cases for both data structures.

**📌 Features**

**Stack (LIFO)**

* Push(T item)
* Pop()
* Peek()
* Clear()
* Count (property)
* Implements IEnumerable<T> for foreach support

**Queue (FIFO)**

* Enqueue(T item)
* Dequeue()
* Peek()
* Clear()
* Count (property)
* Implements IEnumerable<T> for foreach support

**⚠️ Deviations and Assumptions**

* **Raw Array Usage**: Internal storage uses a primitive array (T[]) and resizes by doubling capacity when full.
* **Error Handling**: InvalidOperationException is thrown when Pop, Dequeue, or Peek is called on an empty structure.
* **Iteration Support**: Both classes implement IEnumerable<T> to support foreach iteration.
* **Thread Safety**: The implementation is not thread-safe and assumes single-threaded usage.
* **Generic Support**: Both Stack and Queue are implemented as generic classes (CustomStack<T>, CustomQueue<T>).

**🧪 Example**

var stack = new CustomStack<int>();

stack.Push(10);

stack.Push(20);

Console.WriteLine(stack.Pop()); // Outputs: 20

var queue = new CustomQueue<string>();

queue.Enqueue("Hello");

queue.Enqueue("World");

Console.WriteLine(queue.Dequeue()); // Outputs: Hello