Problem Statement: The "Code-a-Thon" T-Shirt Queue

# Background:

You are a volunteer at a university "Code-a-Thon" event. The first 100 participants to arrive get a free, limited-edition T-shirt. To manage this, you need to create a simple console-based application that tracks the queue of students waiting for their T-shirts.

The process is dynamic: students arrive and join the queue, and sometimes a student has to leave the queue early if they get a call or forget their ID card. The first person in the queue is served when a T-shirt becomes available.

# The Challenge:

Your task is to build a system to manage this queue. The system must handle an **unknown number of students**. You cannot simply reserve space for 100 students in advance, as you don't know how many will show up at any given moment, and the line will constantly change. The system must be efficient when students join or leave the line.

# System Requirements:

You need to implement a program that provides the following options to the event organizer:

1. **Add Student**: A new student arrives and joins the **end** of the queue. The system should ask for the student's name and a unique student ID (an integer).
2. **Serve Student**: The student at the **front** of the queue receives their T-shirt and is removed from the line. The system should display the name of the student who was served. If the queue is empty, it should report that.
3. **Student Leaves**: A student needs to leave the queue unexpectedly. The organizer must be able to remove a student from **any position** in the queue by entering their unique student ID.
4. **Display Queue**: Show the current list of all students in the queue, from front to back, along with their student IDs.
5. **Count Students**: Display the total number of students currently waiting in the queue.

# Data to Store for Each Student:

* Student Name (e.g., "Ali Khan")
* Student ID (e.g., 12345)

# Hints for Your Thought Process (Why this isn't a simple array problem):

* How will you handle the list of students if you don't know the maximum number of people who will join the queue?
* When a student leaves from the middle of the queue, how would you manage the list without having to shift many other students' data around? Think about what would be the most efficient way to "patch up" the line.
* The order of students is crucial. The first one to arrive must be the first one to be served.

# What to Submit:

Submit your source code files. The program should be a menu-driven console application that allows the user to choose from the functions listed above until they choose to exit.

# Sample Console Application:

