Lab Task:

Q1: Write a C++ program to check whether a given number is prime or not. Allow the user to input a number and display whether it's prime or not?

Q2: Design a C++ program to manage student marks. Allow the user to input marks for students in three subjects (Mathematics, English, and Science). The program should calculate the total marks, average marks, and display the grade for each student. The user can specify the number of students and then input the marks for each subject for each student. Finally, display the marks, total, average, and grade for each student. Assume a grading system with the following criteria:

90 or above: Grade A
80-89: Grade B
70-79: Grade C
60-69: Grade D
Below 60: Grade F

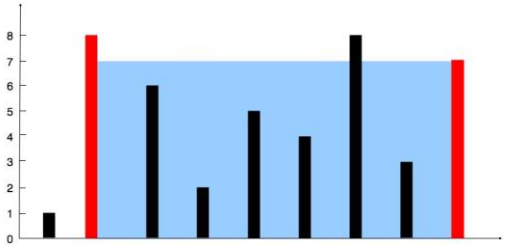
Q3: Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target.

You may assume that each input would have exactly one solution, and you may not use the same element twice.You can return the answer in any order.
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Output: Because nums[0] + nums[1] == 9, we return [0, 1].

Q4: You are given an integer array height of length n. There are n vertical lines drawn such that the two endpoints of the ith line are (i, 0) and (i, height[i]).

Find two lines that together with the x-axis form a container, such that the container contains the most water. Return the maximum amount of water a container can store. Notice that you may not slant the container.

Input: height = [1,8,6,2,5,4,8,3,7]
Output: 49
Explanation: The above vertical lines are represented by array [1,8,6,2,5,4,8,3,7]. In this case, the max area of water (blue section) the container can contain is 49.



Q5: Design Event Management System in C++ to facilitate the organization of events. The program should empower the user to seamlessly manage event details, including the event name, date, venue, and organizer. Incorporate key features such as the ability to add events, display comprehensive details of all events, and perform event searches based on specific dates. To enhance flexibility and scalability, leverage dynamic memory allocation for storing event details.

1. Prompt the user to input the total number of events they wish to manage.
2. Dynamically allocate memory to store event details for the specified number of events.
3. For each event, prompt the user to input:
    - Event Name
    - Date
    - Venue
    - Organizer
4. Display all events that match the provided date, including their complete details.
5. Allow the user to search for events based on a specific date.