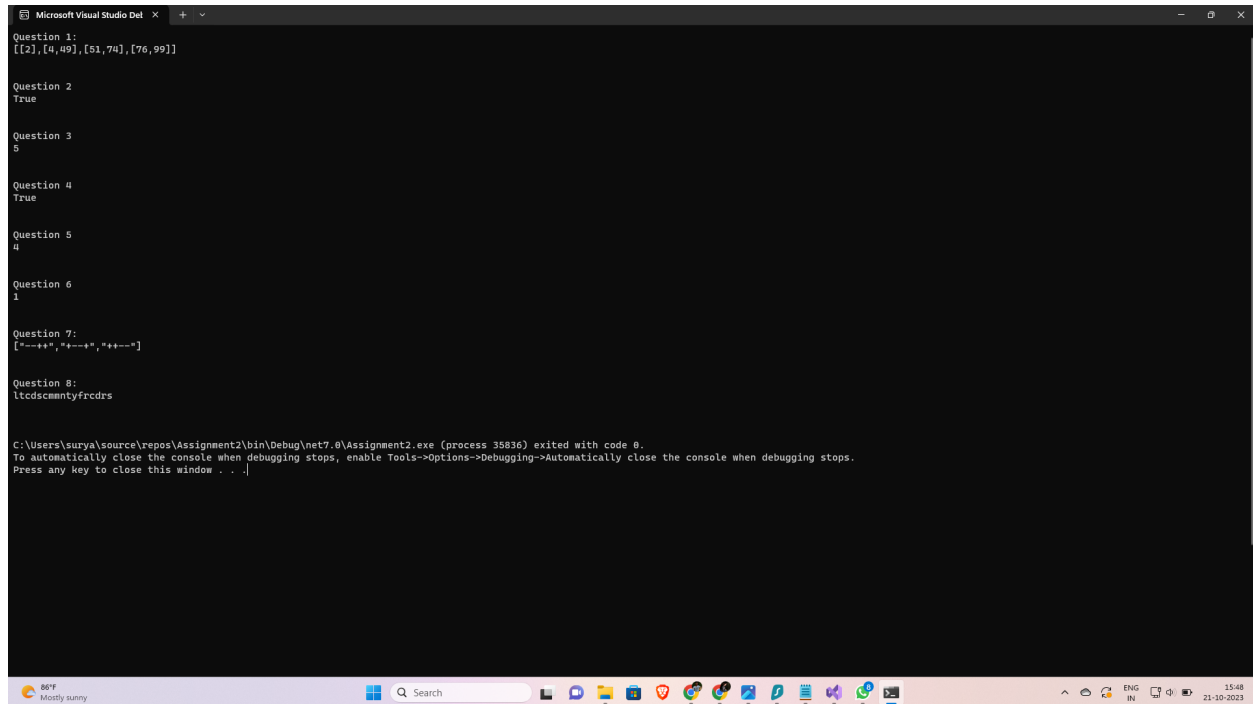


DISTRIBUTED INFORMATION SYSTEM

Assignment - 2

Sai Surya Prakash Munikoti
U42501288

Output Screenshot -



```
Microsoft Visual Studio Det
Question 1:
[[2],[4,49],[51,74],[76,99]]

Question 2
True

Question 3
5

Question 4
True

Question 5
4

Question 6
1

Question 7:
["--++", "--++", "--++"]

Question 8:
ltcdscmmntyfrcdrs

C:\Users\surya\source\repos\Assignment2\bin\Debug\net7.0\Assignment2.exe (process 35836) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Self reflection -

Question 1:-

The task in this question was to find and list the missing ranges within a given inclusive range using a sorted array. This problem highlighted the importance of handling edge cases effectively and writing efficient code. When dealing with large number ranges, it was critical to identify the boundary conditions, particularly the use of long data types to prevent integer overflow. The question emphasized the importance of having a clear understanding of the problem statement in order to implement it correctly. It is critical for future problem-solving to thoroughly understand problem statements, consider edge cases, and rigorously test code to ensure it works effectively in a variety of scenarios.

Question 2 :-

The goal of this question was to determine whether a given string containing brackets was valid or not by applying specific rules. This question demonstrated the significance of employing data structures such as stacks to track open brackets and ensure their proper closure. To deal with this effectively, it was necessary to thoroughly understand the rules and implement a solution

that takes into account different types of brackets, open and close order, and validates their correct usage. The importance of optimizing the algorithm was highlighted by the time complexity constraint. It is critical for future problem-solving to understand the nuances of the problem and apply appropriate data structures and algorithms efficiently, making use of stack structures and working within the given constraints.

Question 3 :-

In this question, the goal was to maximize profit by buying and selling stocks while adhering to certain constraints. This question highlighted the significance of effective algorithm design and problem solving. To accomplish this, the algorithm needed to iterate through the given array of stock prices, keeping track of the lowest price to buy and the highest profit possible. It is possible to maximize profit within the specified time and space constraints by effectively understanding and applying this concept. This question highlighted the importance of iterating through arrays, tracking variables, and leveraging mathematical operations to calculate profits. It's critical to remember the key principles of tracking minima and maximizing profits in such financial scenarios for future problem-solving endeavors.

Question 4 :-

The concept of strobogrammatic numbers, which are numbers that read the same when rotated 180 degrees, was central to this Question. This problem is distinguished by pattern recognition and the comprehension of rotational symmetry in numbers. The solution entailed determining whether the input string was a strobogrammatic number by comparing its characters at corresponding positions while taking their rotational counterparts into account. This question emphasized the importance of recognising patterns and efficiently handling string manipulation within the time and space constraints. When approaching similar problems, it is advantageous to concentrate on rotational symmetry and develop algorithms that account for it, allowing the identification of strobogrammatic numbers.

Question 5 :-

This question delves into the concept of counting good pairs in an array, where a good pair is defined as having the same value with the second element appearing at a higher index than the first. The goal of this problem was to efficiently identify and count pairs of equal elements. The solution involved using a dictionary or hash map to track the frequency of each number and then counting the good pairs by adding the frequency combinations. This question emphasized the importance of understanding data structures and algorithms for efficient counting while also demonstrating how dictionaries or hash maps can help to simplify counting problems. When faced with similar problems, having a solid understanding of data structure operations and combinatorics of counting pairs is critical for developing an efficient algorithm.

Question 6 :-

The goal of this question was to find the third distinct maximum number in an array. To identify distinct maximum numbers, the problem required sorting the array in descending order. It was critical to understand how to handle situations with fewer than three distinct maximums. Sorting the array, removing duplicate maximums, and selecting the third distinct maximum were all part of the solution. If there were fewer than three distinct maximums, the maximum was used as a fallback. This question emphasized the importance of array sorting, dealing with duplicates, and selecting specific elements from an array. It also stressed the significance of comprehending array sorting algorithms and their complexities. A solid understanding of array manipulation, sorting, and condition handling is required when approaching similar problems.

Question 7 :-

Question 7 asked me to generate all possible game states after a valid move using a game scenario involving consecutive character flips in a given string. It took a methodical approach to efficiently iterate through the string, identify valid moves, flip the characters, and assemble the resulting states. While this problem primarily tested string manipulation and logical reasoning, it also demonstrated the significance of developing algorithms for games and simulations. The solution emphasized the importance of effective string handling, game state management, and rule adherence, emphasizing the necessary skills for tackling similar programming challenges.

Question 8 :-

Question 8 involved removing specific vowels from a string and returning the modified string. This task demonstrated the value of string manipulation and pattern matching in text processing. The proposed solution used a HashSet to efficiently store the vowels to be removed before iterating through the input string and filtering out the unwanted characters. The exercise emphasized the importance of effectively handling text data and demonstrated how fundamental data structure knowledge can be applied in real-world scenarios. Solving such issues is critical for tasks like data cleaning, text analysis, and text transformation, which are common in software development.