Lab - Activity - 1

Instructions:

1. Complete all the tasks below.
2. Write clean, well-documented code for each task.
3. The code needs to be written in any IDE of your choice, compiled, and successfully executed.
4. Discuss any challenges you faced and how you overcame them. Compilation errors, runtime errors, etc., pics of each error should go into the report.

---

1. You are given a set of n activities with start times s_i and finish times f_i. Implement a C++ program using the Greedy Strategy to select the maximum number of non-overlapping activities.
2. A digital archivist is tasked with preserving a massive collection of ancient manuscripts by digitizing them into a single compressed file. To minimize the amount of physical memory used, the archivist decides to use a variable-length encoding scheme based on the frequency of character occurrences within the text. Characters that appear frequently should be represented by shorter binary codes, while rare characters receive longer ones. Implement a program that calculates the frequency of each unique character in a given paragraph and constructs a Huffman Tree using a min-priority queue to generate the optimal prefix codes.
3. A financial analyst is examining the historical daily price changes of a specific stock over a period of n days. These changes are recorded as an array of integers, where positive values represent gains and negative values represent losses. The analyst needs to identify the "golden period"—a contiguous sequence of days during which the sum of price changes is the highest—to determine when an investor would have made the most profit. Implement a recursive Divide and Conquer algorithm that splits the array into halves to find the maximum sum entirely in the left, entirely in the right, or crossing the midpoint.
4. A security engineer is developing a new encryption protocol that requires the multiplication of exceptionally large prime numbers, often exceeding 100 digits each. Using the traditional multiplication method would be too slow, potentially leading to a "75-year execution time" for extremely large datasets8. To ensure the system is responsive, the engineer implements the Karatsuba algorithm to reduce the number of recursive multiplications required. Create a program that multiplies two n-digit integers by splitting them into high and low components and recursively calculating only three sub-products.