Assignment - 43 **A Job Ready Bootcamp in C++, DSA and IOT**

priority_queue

1. Write a c++ program, to demonstrate priority queue.
2. Implement different operations on priority queue .i.e. adding element, removing element, size of priority queue, and print it.
3. Write a c++ program, to demonstrate priority queue having a min element at top.
4. Write a c++ program, to swap the elements of two priority queues of int type.
5. Write a c++ program, to show that priority_queue is by default a Max Heap.
   Note:
   If elements are printed in descending order, then we have a max heap.
6. Write a c++ program, to use priority_queue to implement min heap.
7. Given two sorted arrays A[] and B[] of sizes N and M respectively, the task is to merge them in a sorted manner using priority_queue.
   Example:
   Input: A[] = { 5, 6, 8 }, B[] = { 4, 7, 8 }
   Output: 4 5 6 7 8 8
8. Given an array arr[] of N elements, the task is to perform using priority_queue and the following operation:

- Pick the two largest element from the array and remove these element. If the elements are unequal then insert the absolute difference of the elements into the array.
- Perform the above operations until the array has 1 or no element in it. If the array has only one element left then print that element, else print “-1”.
  Example:
  Input: arr[] = { 3, 5, 2, 7 }
  Output: 1
  Explanation:
  The two largest elements are 7 and 5. Discard them. Since both are not equal, insert 7 – 5 = 2 into the array. Hence, arr[] = { 3, 2, 2 }
  The two largest elements are 3 and 2. Discard them. Since both are not equal, insert 3 – 2 = 1 into the array. Hence, arr[] = { 1, 2 }
  The two largest elements are 2 and 1. Discard them. Since both are not equal, insert 2 – 1 = 1 into the array. Hence, arr[] = { 1 }
  The only element left is 1. Print the value of the only element left.

9. Given three arrays X[], Y[], and Z[] each consisting of N integers, the task is to find the maximum number of triplets (X[i], Y[i], Z[i]) such that (X[i] < Y[i] < Z[i]) for any permutation of the three arrays using priority_queue
   Input: X = {9, 6, 14, 1, 8}, Y = {2, 10, 3, 12, 11}, Z = {15, 13, 5, 7, 4}
   Output: 3
   Explanation:
   After rearranging the arrays X[], Y[] and Z[] as {1, 6, 8, 9, 14}, {3, 2, 10, 12, 11}, and {4, 7, 15, 13, 5} respectively. The increasing triplets are {1, 3, 4}, {8, 10, 15} and {9, 12, 13}.
   Therefore, the total count of such triplets is 3.

10. Given an array arr[] of size N and a number K, the task is to find the length of the smallest subsequence such that the sum of the subsequence is greater than or equal to number K do it using priority_queue.
    Example:
    Input: arr[] = {2, 3, 1, 5, 6, 3, 7, 9, 14, 10, 2, 5}, K = 35
    Output: 4
    Smallest subsequence with the sum greater than or equal to the given sum K is {7, 9, 14, 10}
    Input: arr[] = {1, 2, 2, 2, 3, 4, 5, 4, 7, 6, 5, 12}, K = 70
    Output:-1
    Subsequence with sum greater than equal to the given sum is not possible.