1. Write a C++ program to demonstrate the addition of multiple types of data using generic functions or templates.

2. Write a C++ Program to find Largest among two numbers using function template.

3. Write a C++ program to find the largest of three elements using a template.

4. Write a C++ Program to Swap data using function template.

5. Write a C++ Program to Add two numbers using function template.

6. Write a C++ Program to find Sum of Array using function template.

7. Write a C++ Program of Templated class derived from Non-templated class.

8. Write a C++ Program to implement push and pop methods from stack using template.

9. Write a C++ Program to Perform Simple Addition Function Using Templates.

10. Write a C++ program to implement Hash Table using Template Class.

Assignment - 36
STL: array

1. Using STL Array gets and sets a reference to an element based on a given index.
2. Using STL Array returns the total number of elements in the array.
3. Find the first and last element using the STL array.
4. Returns the element from the given index using the STL array.
5. C++ STL program to demonstrate example of array::rbegin() and array::rend()
functions

6. Using STL to check whether an array is empty or not.
7. Sort an array in ascending order using sort() function in C++ STL
8. Sort an array in descending order using sort() function in C++ STL
9. C++ program to find the integers which come an odd number of times in an array using C++ STL.
10. Given an integer array nums , return an array answer such that answer[i] is equal to the product of all the elements of nums except nums[i] .

Assignment - 37
STL: vector

1. Declare a vector with Initialization and print the elements.
2. Declare a vector without initialization, insert some elements and print
3. Write a function to print the element of a vector and take input from the user.
4. Write a program to Copy one vector's elements to another vector.

5. Find largest and smallest elements in a vector
6. Write a program to reverse vector elements
7. Write a program to find sum of vector elements
8. Write a program to find common elements between two vectors.
9. Write a program to Push and print elements in a float vector
10. Write a program to check whether an element exists in a vector or not.

Assignment - 38
list

1. List functions in C++ STL (Standard Template Library)
2. Assign the elements to the list (different methods) - Example of list::assign() | C++ STL
3. Iterate a list C++ STL
4. Iterate a list in reverse order C++ STL
5. Input and add elements to a list C++ STL
6. Get the first and last element of the list C++ STL
7. Insert the element at beginning and end of the list | C++ STL
8. Remove all occurrences of an element and remove set of

some specific from the list
C++ STL
9. Remove all consecutive duplicate elements from the list |
C++ STL
10. Merge two lists C++ STL
11. Creating a list by assigning the all elements of another list
C++ STL
12. Assign a list with array elements C++ STL
13. Push characters in a list and print them separated by
space in C++ STL
14. Access elements of a characters list using const_iterator
in C++ STL

Assignment - 39
forward_list

1. Write a c++ code, to demonstrate the forward list.
2. Write a c++ code, in which create a forward list and assign
values to it at the time of
initialization and print it on the console screen.
3. Create a forward list insert elements from 1 to 10 and find
the sum of elements.
4. Write a program to reverse forward list elements.

5. Write a program remove all consecutive duplicate elements from the forward list

6. Create two forward lists of int type, and merge them.

7. Below are two forward lists, first sort them and then merge them.

forwardlist1={3,2,9}

forwardlist2={8,1,2}

8. Create two forward lists of int type, and swap the elements of both forward lists with

each other.

9. Write a C++ code to demonstrate working of splice_after() in forward list.

10. Write a program to assign values in forward_list using the values of another list

Assignment-40: deque

1. Inserts an element. And returns an iterator that points to the first of the newly inserted

elements.

2. Returns a reverse iterator which points to the last element of the deque (i.e., its

reverse beginning).

3. Returns a reverse iterator which points to the position before the beginning of the
deque (which is considered its reverse end).

4. Returns a constant iterator pointing to the first element of the container, that is, the
iterator cannot be used to modify, only traverse the deque.

5. Returns the maximum number of elements that a deque container can hold.

6. Assign values to the same or different deque container.

7. Return the first element and last element of the deque container.

8. Remove elements from a container from the specified position or range in deque.

9. Generate a permutation of first N natural numbers having count of unique adjacent

differences equal to K | Set 2 using a queue.

10. Check if given Strings can be made equal by inserting at most 1 String using deque.

11. How to get the first and last elements of Deque in c++?

12. Given a string S containing letters and '#'. The '#" represents a backspace. The task
is to print the new string without '#'. String after processing backspace characters
using deque
Examples:
Input : S = "abc#de#f#ghi#jklmn#op#"
Output : abdghjklmo
Input : S = "##iNeuron##Education##hub#"
Output : iNeurEducatihu

13. Segregate even and odd nodes in a Linked List using Deque.

Assignment - 41
Stack

1. Create a stack of int type, push 5 elements in it and print it on the console screen.

2. Create a stack of int type, and find the top most element in a stack.

3. Create a stack, and implement main operations like push(), pop(), peek(), empty()
and size().

4. Reverse the Words of a String using Stack.
Example:
Input: str = "I Love To Code"
Output: Code To Love I

5. Create stack1 of int type, and create another stack of the same type with name
stack2 and copy all the elements of stack1 into stack2 in the same order.

6. Reverse a string using a stack.

Example:

Input: str = "Reverse me"

Output: em esreveR

7. Create a stack of int type and sort it.

8. Create a stack of int type and sort it in descending order.

9. Create back button functionality using stack.

10. Given an array, print the Next Greater Element (NGE) for every element using stack.

Example:

Input: arr[] = [4 , 5 , 2 , 25]

Output: 4 –> 5

5 –> 25

2 –> 25

25 –> -1

Assignment - 42
Queue

1. Print all elements of a queue in C++ STL
2. Create a railways ticket counter and allot ticket to first come first take using c++ STL
3. Check if a queue is empty or not using queue::size() function
4. Manage a queue for multiple input and store in ascending order of his first character
5. Schedule a interview by using applicant's reaching time using c++ STL
6. In C++ STL, Queue is a type of container that follows FIFO (First-in-First-Out)
elements arrangement i.e. the elements which insert first will be removed first. In the
queue, elements are inserted at one end known as "back" and are deleted from
another end known as "front".empty() and size() function of the queue with the
Example.
7. Exchange the contents of two queues but the queues must be of the same data type,

although sizes may differ.
8. Insert a new element into the queue container, the new element is added to the end
of the queue
9. Adds the element 'g' at the end of the queue.
10. Deletes the first element of the queue.

Assignment - 43
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.

Assignment - 44
set

1. Find the total number of elements of the set container.

2. Using inbuilt function and insert an element to the set container.

3. Erase an element from a set.

4. Checks whether the set is empty or not.if it is empty insert 5 elements in the set

5. Make a c++ program to insert unique element in set

6. How to find the lower bound of any desired element from the set.

7. how to find the upper bound of any desired element from the set.

8. Create a function to search the element in the set.

9. Converting String into Set in C++ STL

10. You have a set of integer s, which originally contains all the numbers from 1 to n.
Unfortunately, due to some error, one of the numbers in s got duplicated to another
number in the set, which results in repetition of one number and loss of another
number.You are given an integer array nums representing the data status of this set
after the error.Find the number that occurs twice and the number that is missing and
return them in the form of an array.

Assignment - 45 multiset

1. Create a c++ program using multiset and returns an iterator to the first element in
the multiset –> O(1)

2. Create a c++ program using multiset and returns an iterator to the theoretical
element that follows the last element in the multiset –> O(1)

3. Create a c++ program using multiset and returns the number of elements in the
multiset –> O(1)

4. Create a c++ program using multiset and returns the maximum number of elements
that the multiset can hold –> O(1)

5. Create a c++ program using multiset and returns whether the multiset is empty –>
O(1)

6. Create a c++ program using multiset and inserts the

element x in the multiset –>
O(log n)

7. Create a c++ program using multiset and removes all the elements from the multiset
–> O(n)

8. Create a c++ program using multiset and removes all the occurrences of x –> O(log
n)

9. Create a c++ program using multiset and remove only one instance of element from
multiset having same value

10. Unlike a set, a multiset may contain multiple occurrences of same number. The
multiset equivalence problem states to check if two given multisets are equal or not.
For example let A = {1, 2, 3} and B = {1, 1, 2, 3}. Here A is set but B is not (1 occurs
twice in B), whereas A and B are both multisets. More formally, "Are the sets of pairs

defined as $A' = \{ (a, frequency(a)) \mid a \in \mathbf{A} \}$ equal for the two given

multisets?" Given two multisets A and B, write a program to check if the two multisets

are equal.

Assignment-46:

map

1. Create a map, insert at least 5 pairs of keys and values and print it.

2. Create a map, where insert keys and values as string type and integer type

respectively and print it on the screen.

3. Create a map, insert some pairs and print all elements in reverse order using rbegin

and rend function.

4. Create a map, and insert some pairs and find one pair out of the inserted pair and
replace it with another pair and print map.

5. Create a map, insert some pairs and Find the occurrence of each pair and print it on
the screen.

6. Create a map, use a member function to tell whether a map is empty or not and then
insert some pairs into the map and find the size of map.

7. Sort a given map in descending order based on values instead of keys in C++ STL.
Key value
1 6
2 8
6 3
8 2

8. Given a string s of length N, containing digits written in words but in jumbled form, the
task is to find out the digits present in the string in word form

and arrange them in
sorted order using map
Example:
Input: s = "ozerotwneozero"
Output: 0012
Explanation: The string can be arranged as "zerozeroonetwo".
Therefore the digits are 0, 0, 1 and 2.

9. Given two maps map1 and map2 having a string as the key
and arrays of integers as
values, the task is to merge them in one map such that if a
key is common in both the
maps, the respective arrays should be merged.
Example:
Input: map1 = { ("key1", {0, 1}), ("key2", {0, 1}) }, map2 = {
("key2", {1, 2}) };
Output: { (key1, {0, 1}), (key2, {0, 1, 2}) }
Explanation: After merging key1 array will become {0, 1} and
for key2 after merging
array will become {0, 1, 2}

10. Given a positive integer N, the task is to check whether N
can be represented as the

difference between two positive perfect cubes or not. If found to be true, then print

"Yes". Otherwise, print "No" using a map.

Example:

Input: N = 124

Output: Yes

Explanation: Since 124 can be represented as (125 – 1) = (53 – 13). Therefore, print

Yes.

Assignment-47: Pair and Tuple

1. Given an array of pairs arr[] of size N (N ≥ 3) where each element of pair is at most N
and each pair is unique, the task is to determine the number of ways to select triplets
from the given N pairs that satisfy at least one of the following conditions:
   1. The first value (a) of each pair should be distinct.
   2. The second value (b) of each pair should be distinct.

2. Sorting Vector of Pairs by 1st element in ascending and 2nd

element in descending.

3. Create a C++ program to implement the deque of pairs.

4. Given an array of pairs A[][] of size N, the task is to find the longest subsequences
where the first element is increasing and the second element is decreasing.
Examples:
Input: A[]={{1, 2}, {2, 2}, {3, 1}}, N = 3
Output: 2
Explanation: The longest subsequence satisfying the conditions is of length 2 and
consists of {1, 2} and {3, 1};
Input: A[] = {{1, 3}, {2, 5}, {3, 2}, {5, 2}, {4, 1}}, N = 5
Output: 3

5. Given an array arr[] consisting of N integers and an integer X, the task is to perform
integer division on the array elements by X and print the indices of the array in
non-decreasing order of their quotients obtained.
Examples:

Input: N = 3, X = 3, order[] = {2, 7, 4}
Output: 1 3 2
Explanation:
After dividing the array elements by 3, the array modifies to {0, 2, 1}. Therefore,
the required order of output is 1 3 2.
Input: N = 5, X = 6, order[] = {9, 10, 4, 7, 2}
Output: 3 5 1 2 4
Explanation:
After dividing the array elements by 6, the array elements modify to 1 1 0 1 0.
Therefore, the required sequence is 3 5 1 2 4.

6. Implementation of lower_bound() and upper_bound() in List of Pairs in C++

7. Implementation of lower_bound() and upper_bound() in Array of Pairs in C++

8. Priority queue of pairs in C++ with ordering by first and second element

9. Check if a given pair of Numbers are Betrothed numbers or

not

10. Sort an Array of Points by their distance from a reference Point

11. create a tuple and demonstrate all the inbuilt functions of tuples

12. Create an unordered_map of tuples in C++?

13. Iterate over the elements of an std::tuple in C++

14. Sorting of Vector of Tuple in C++ (Descending Order)

15. Sorting of Vector of Tuple in C++ (Ascending Order)

16. Create a C++ program to demonstrate the working of forward list of tuples.

17. Create a C++ program to implement max-heap priority queues of tuples.

Assignment-48:

multimap

1. Write a c++ program to demonstrate the implementation of multimap

2. Declare a multimap m1 of key-value pairs of integer type and then insert some pair
type data. Now print the multimap values of m1, and also create another multimap
m2 of the same type as m1 using m1.begin() and m1.end() as parameters.

3. Write a c++ code for illustration of multimap::swap() function.

4. Write a program to erase all the entries of the key.

5. Write a program to erase only a single value based on position.

6. Write a program to find some key value pairs and print on

the console.

7. Write a program to find a lower bound.

8. Write a program to find the upper bound.

9. You are given an array A of size N. You need to insert the elements of A into a
multimap(element as key and index as value) and display the results. Also, you need
to erase a given element x from the multimap and print "erased x" if successfully
erased, else print "not found".

10. Write a program that checks whether a given multimap is empty or not.