OBJECT ORIENTED PROGRAMMING I LABORATORY
Experiment # 9:
STL I

QUESTIONS

1) Write a C++ program to implement a music playlist. Create a **linked-list** as in type **string**.

- Add two songs at beginning of the list and add two songs at end of the list. Print the playlist.
- Sort the playlist. Print the playlist.
- Define a string array and initialize it with four songs. Also create a second **linked-list** as in type **string** and initialize that playlist by using the string array. Print the playlist.
- Remove second playlist elements and insert at end of first playlist (**Hint**: Use **splice** function). Print both of the playlist.
- Sort the first playlist. Print the first playlist.
- Insert elements of array into the second playlist. Sort the second playlist. Print the second playlist.
- Remove second playlist elements and insert into the first playlist in sorted order (**Hint**: Use **merge** function). Print both of the playlist.
- Remove the first and last elements of the first playlist. Print the first playlist.
- Remove duplicate elements from the first playlist. Print the first playlist. (**Hint**: Use **unique** function).
- Swap elements of the first and the second playlists. Print both of the playlist. (**Hint**: Use **swap** function).
- Replace contents of playlist with elements of second playlist. Print both of the playlist. (**Hint**: Use **assign** function).
- Remove a song from the first playlist by using **remove** function. Print the first playlist after remove operation.

2) Consider we have a basketball team. The following table shows the number of the players and points of each of player. Write a C++ program to analyze the performance of the team. Use **map** to store number and points of the players.

Number	Points
0	20
12	4
18	9
21	4
22	24
23	8
42	4
44	8

- Insert the data given below to the map. Print the map.
- Find the player 12 and prints his points.
- Find upper bound of 23 and lower bound of 18.
- Update points of 44 with 12. Print the map after update operation.
- Add a new player with number 1 and points 10. Print the map after add operation.

E-mail : burakaleci@gmail.com

3) Write a C++ program to reverse items of the data by using **stack** data structure. An example for reversing operation is given below:

Data: 2 8 3 6 1 5 4 9 0 7
Reversed data: 7 0 9 4 5 1 6 3 8 2

- Define an array that stores 10 integer numbers (i.e. numbers[10]), then initialize the array with the given data.
- Use **stack** data structure to reverse the array.
- Assign reversed data to the array.
- Print the array.

4) Write a class sortingAlgorithms that includes Bubble Sort, Selection Sort, Insertion Sort, Quick Sort, Merge Sort, Shell Sort, Heap Sort sorting algorithms. These algorithms must sort the integer vectors in ascending order. Also execute sort algorithm in STL.

i. Generate a random vector with 100 items
ii. Run all sorting algorithms for that vector and measure the elapsed time for sorting process.
iii. Repeat i and ii for 1000, 10000, 100000, 1000000, 1000000 items.
iv. Draw a chart to demonstrate the time performance of the algorithms depending on number of items.
v. Consider worst-case time complexities of the algorithms and discuss the results.
(Hint: You can get help to implement algorithms from the web site: https://www.softwaretestinghelp.com/sorting-techniques-in-cpp/)