Assignment No :- 9

Aim :-

a) Write C++ program using STL to add binary numbers (assume one bit as one number); use STL stack

b) Write C++ program using STL map for managing person Record. (Name, birth date, telephone no.) Perform operations - add, display, search, delete, and update.

Theory :-

1. Explain following standard Template Library (STL) component in details.

a) Algorithms :-
The header algorithm defines a collections of functions especially designed to be used on ranges of elements. They act on containers & provide means for various operations for the contents of the containers

b) Containers :-
Containers or container classes store objects & data. There are in total seven standard "first-class" container classes & three container adaptor classes & only seven header files that provide access to these containes or container adaptors.

c) Functions :-
The STL includes classes that overload the functions call operator. Instances of such classes are called function objects or functors. Functors allow the working of the associated functions to be customized with the help of parameters to be passed.

d) Iterators
As the name suggests, iterators are used for working upon a sequence of values. They are the major feature that allow generality in STL.

Conclusion :-
We have succesfully implemented the concept of STL and we studied different members of STL.

CODE:

```
/*
NAME:SARVESH BAPUSAHEB CHAVAN
ROLL NO:SYCOA124
*/
#include <iostream>
#include <string>
#include <map>
using namespace std;
int main()
{
    map<string, int> Person;
    int choice, t;
    char name[20];
    while (1)
    {
        cout << "\nMENU" << endl;
        cout << "1. Insert" << endl;
        cout << "2. Search" << endl;
        cout << "3. Display" << endl;
        cout << "4. Delete" << endl;
        cout << "5. Exit" << endl;
        cout << "Enter choice: ";
        cin >> choice;
        switch (choice)
        {
        case 1:
        {
            cout << "Name: ";
            cin >> name;
            cout << "Telephone no.: ";
            cin >> t;
            Person.insert(std::pair<string, int>(name, t));
            break;
        }
        case 2:
        {
            cout << "Enter name: ";
            cin >> name;
            std::map<string, int>::iterator it = Person.find(name);
            if (it != Person.end())
                std::cout << "Telephone no.: " << Person.find(name)->second <<
'\n';
            break;
        }
        case 3:
        {
```

```
            for (map<string, int>::iterator it = Person.begin(); it !=
Person.end(); ++it)
                cout << (*it).first << "  " << (*it).second << endl;
            break;
        }
        case 4:
        {
            cout << "Enter name: ";
            cin >> name;
            map<std::string, int>::iterator it = Person.find(name);
            Person.erase(it);
            break;
        }
        case 5:
            return 0;
        default:
            cout << "Wrong input\n";
        }
    }
    return 0;
}
```

OUTPUT:

```
MENU
1. Insert
2. Search
3. Display
4. Delete
5. Exit
Enter choice: 1
Name: sarvesh
Telephone no.: 1452

MENU
1. Insert
2. Search
3. Display
4. Delete
5. Exit
Enter choice: 1
Name: lomesh
Telephone no.: 1426

MENU
1. Insert
2. Search
3. Display
4. Delete
5. Exit
Enter choice: 2
Enter name: lomesh
Telephone no.: 1426
```

```
MENU
1. Insert
2. Search
3. Display
4. Delete
5. Exit
Enter choice: 3
lomesh  1426
sarvesh  1452

MENU
1. Insert
2. Search
3. Display
4. Delete
5. Exit
Enter choice: 4
Enter name: lomesh

MENU
1. Insert
2. Search
3. Display
4. Delete
5. Exit
Enter choice: 3
sarvesh  1452

MENU
1. Insert
2. Search
3. Display
4. Delete
5. Exit
Enter choice: 5
PS D:\program\secondyear>
```