Page:
Date: / /

Assignment 12

Class - SE IV   Roll NO - 21430.
Batch - F4
D.O.S . 27/11/2020

Problem statement :-
Write C++ program to map container (associate)
The Keys will be the names of States and the values will be the population of the states. When the program runs the user is promoted to type the name of state. The program then looks in the map using the state name as an index and returns the population of the States.

Objectives :-
To learn the concept of associative Container.

Theory :-
Map associative Container are associate container that store elements in a mapped fashion Each element has a key value and a mapped value. No two mapped values can be have same key values.
map :: operator [ ] -
This operator is used to reference the element present at position given inside the operator. It is similar to the at( ) function.

The only difference is that at() function throws an out of range exception when the position is nothing the bounds of the sizeof map while this operator causes undefined behaviour.

Syntax:

mapname[key]
parameters.
key value mapped to the element to be fetched
Returns-
Direct refereno to the element at given key value

E.g.
```
int main()
{  map < int, string>mymap;
   mymap [1] = "Hi";
   mymap [2] = "This";

   cout<<mymap[2] ;    //This.

}
```

Algorithm-:
1. Start
2. Give a header file to map associative container.
3. Insert states names so that we get values as population of that state.

Page:
Date: / /

4. Use population.insert()
5. Display the population of states.
6. END.

* Pseudocode-:

```
int main()
{
    1. map <string,int>st;
    2. map<string, int>:: iterator j=st.begin():

    3. Define int ch,pp.flag;
    4. Define string nm;
    5. DO
      {
          Display "Menu"
                    1. Insert
                    2. find
                    3. Display
                    4. Exit.
          Read choice
          switch (choice)
          {
              case 1:
              Read state name and population
              st.insert <pair<string,int>(nm,pp));
              break;
```

Page:
Date: / /

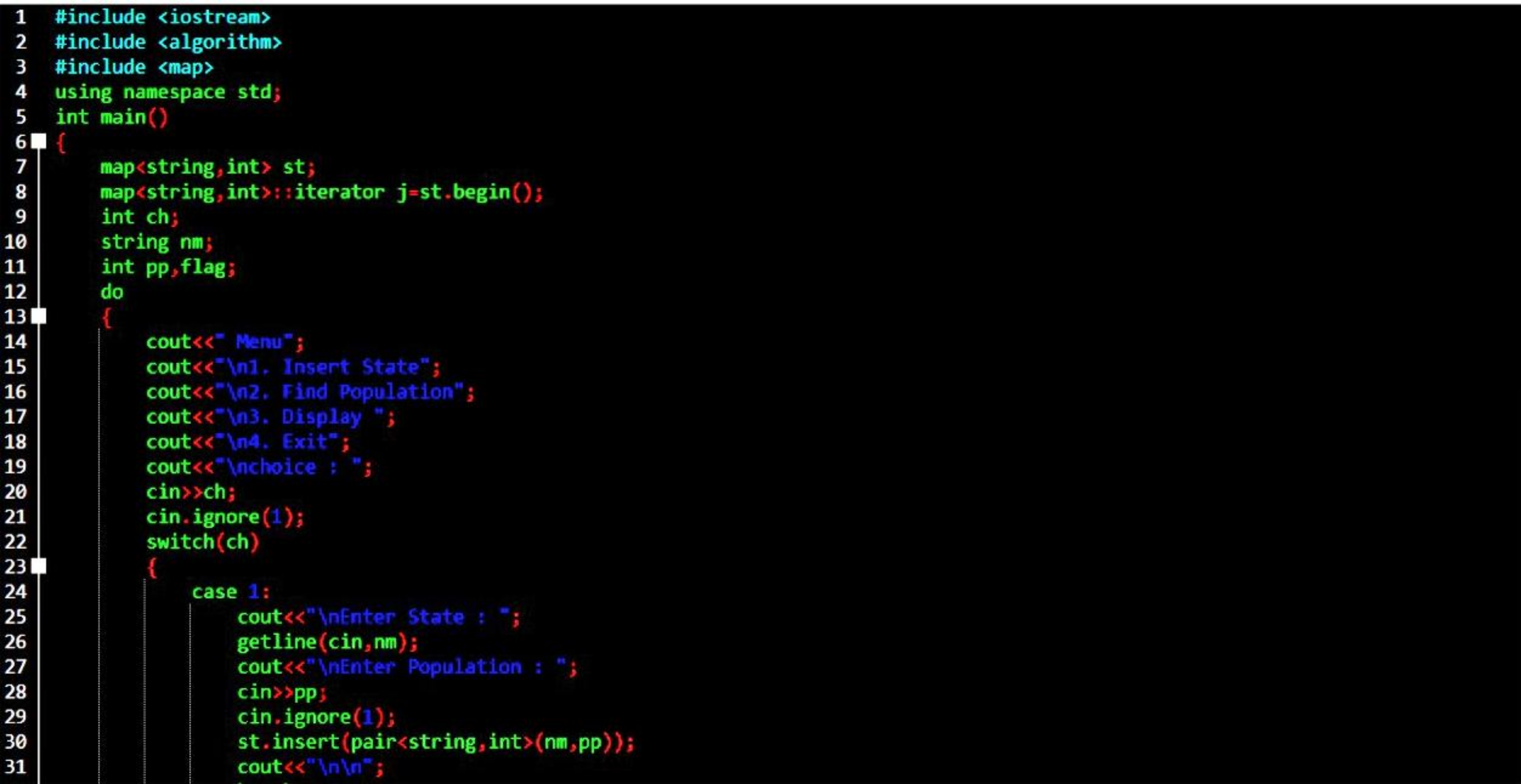
```
        case 2 :
            flag = 0
            Read state(nm)
            j = st.begin()
            for ( j=st.begin() ; j!=st.end() ; j++)
            {
                if ( nm==j→first)
                {
                    flag 1 ;
                    Display state(nm) & population(pp)
                }
            }
            if (flag=~~=~~0)
            {
                Display "Not found";
            }
        case 3:
            j = st.begin()
            While ( j! = st.end() )
            {
                Display nm, pp;
                j++;
            }
            break;
        case 4:
            Display "Thank You"
            Exit(1)
    }
} While (ch);
END.
```

Page:
Date: / /

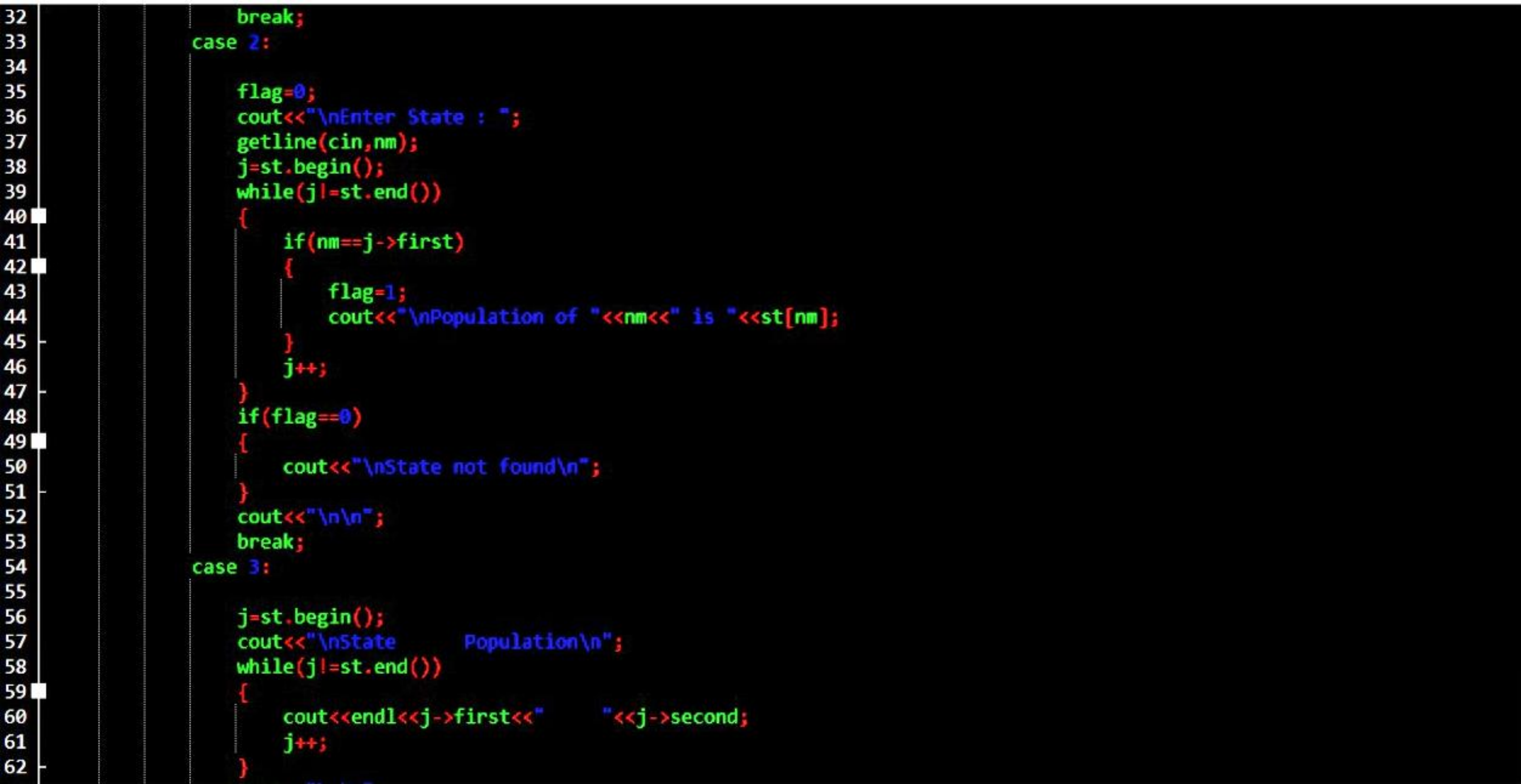
Test cases.

NO.	Description	Input	Expected o/p	Actual o/p	Result.
1.	Menu-1 1. Insert 2. Find 3. Display 4. Exit.	ch = 1 st - Mah Pp - 22,500	st - Mah pp - 22,500	st - Mah pp - 22,500	pass
2.	Menu 1. Insert 2. Find 3. Display 4. Exit	ch = 2 st = Delhi	Not found	Not found	pass

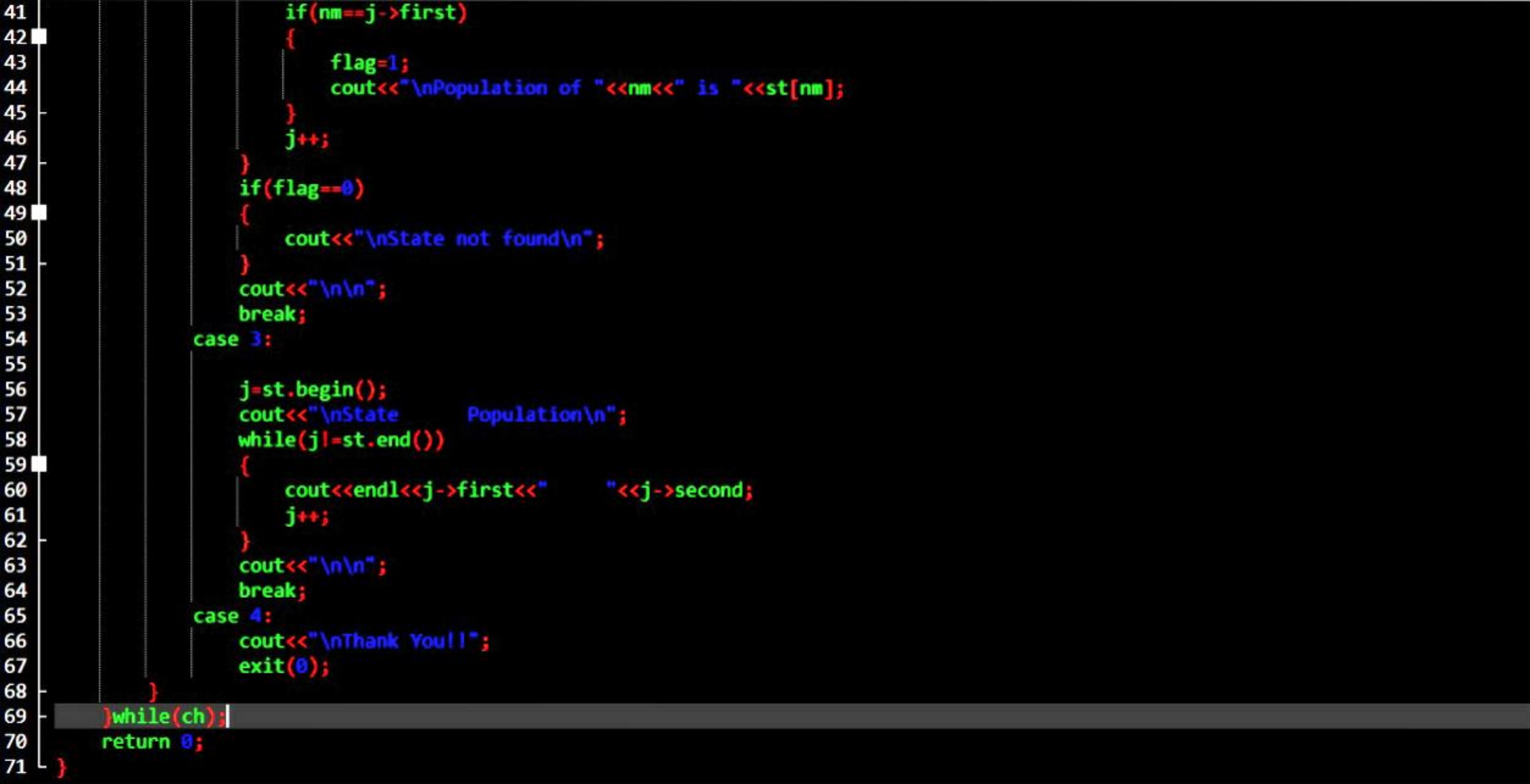
Conclusion-:
We successfully studied and implement concept of map associative container.



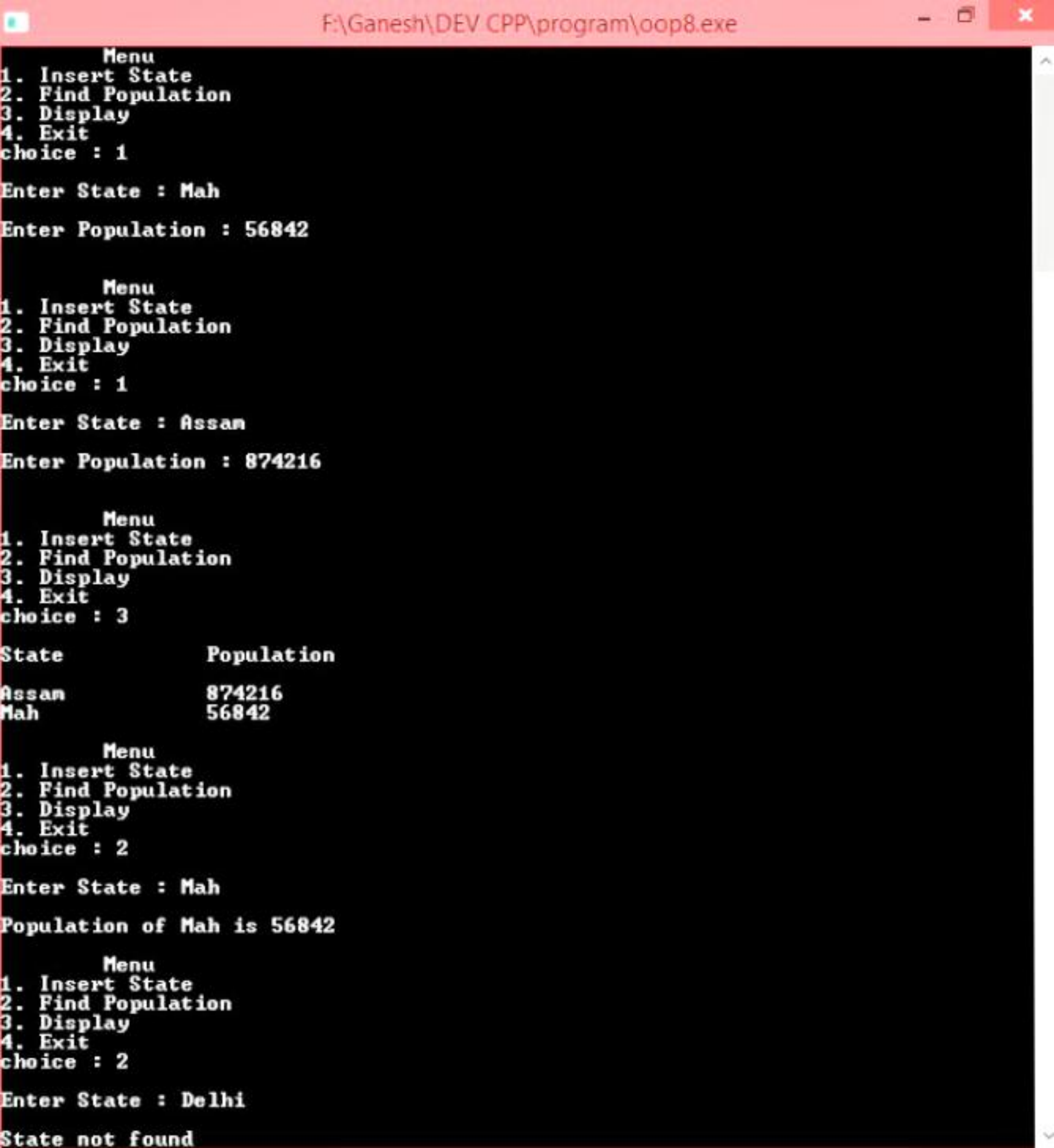
```cpp
#include <iostream>
#include <algorithm>
#include <map>
using namespace std;
int main()
{
    map<string,int> st;
    map<string,int>::iterator j=st.begin();
    int ch;
    string nm;
    int pp,flag;
    do
    {
        cout<<" Menu";
        cout<<"\n1. Insert State";
        cout<<"\n2. Find Population";
        cout<<"\n3. Display ";
        cout<<"\n4. Exit";
        cout<<"\nchoice : ";
        cin>>ch;
        cin.ignore(1);
        switch(ch)
        {
            case 1:
                cout<<"\nEnter State : ";
                getline(cin,nm);
                cout<<"\nEnter Population : ";
                cin>>pp;
                cin.ignore(1);
                st.insert(pair<string,int>(nm,pp));
                cout<<"\n\n";
```



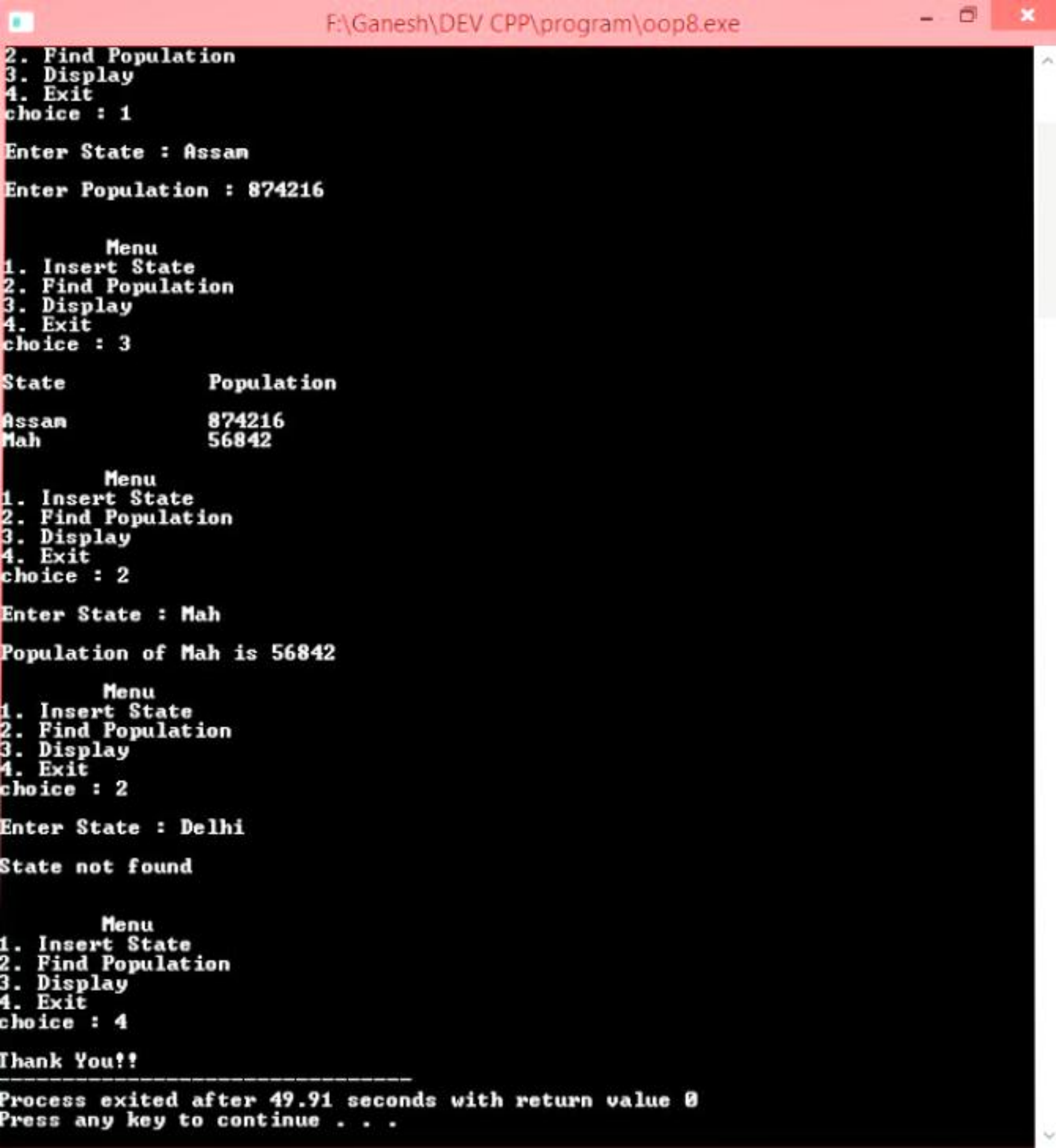
```
                break;
            case 2:

                flag=0;
                cout<<"\nEnter State : ";
                getline(cin,nm);
                j=st.begin();
                while(j!=st.end())
                {
                    if(nm==j->first)
                    {
                        flag=1;
                        cout<<"\nPopulation of "<<nm<<" is "<<st[nm];
                    }
                    j++;
                }
                if(flag==0)
                {
                    cout<<"\nState not found\n";
                }
                cout<<"\n\n";
                break;
            case 3:

                j=st.begin();
                cout<<"\nState      Population\n";
                while(j!=st.end())
                {
                    cout<<endl<<j->first<<"      "<<j->second;
                    j++;
                }
```



```cpp
                    if(nm==j->first)
                    {
                        flag=1;
                        cout<<"\nPopulation of "<<nm<<" is "<<st[nm];
                    }
                    j++;
                }
                if(flag==0)
                {
                    cout<<"\nState not found\n";
                }
                cout<<"\n\n";
                break;
            case 3:

                j=st.begin();
                cout<<"\nState      Population\n";
                while(j!=st.end())
                {
                    cout<<endl<<j->first<<"     "<<j->second;
                    j++;
                }
                cout<<"\n\n";
                break;
            case 4:
                cout<<"\nThank You!!";
                exit(0);
        }
    }while(ch);
    return 0;
}
```



```
        Menu
1. Insert State
2. Find Population
3. Display
4. Exit
choice : 1

Enter State : Mah

Enter Population : 56842


        Menu
1. Insert State
2. Find Population
3. Display
4. Exit
choice : 1

Enter State : Assam

Enter Population : 874216


        Menu
1. Insert State
2. Find Population
3. Display
4. Exit
choice : 3

State           Population

Assam           874216
Mah             56842

        Menu
1. Insert State
2. Find Population
3. Display
4. Exit
choice : 2

Enter State : Mah

Population of Mah is 56842

        Menu
1. Insert State
2. Find Population
3. Display
4. Exit
choice : 2

Enter State : Delhi

State not found
```



```
2. Find Population
3. Display
4. Exit
choice : 1

Enter State : Assam

Enter Population : 874216


        Menu
1. Insert State
2. Find Population
3. Display
4. Exit
choice : 3

State               Population

Assam               874216
Mah                 56842

        Menu
1. Insert State
2. Find Population
3. Display
4. Exit
choice : 2

Enter State : Mah

Population of Mah is 56842

        Menu
1. Insert State
2. Find Population
3. Display
4. Exit
choice : 2

Enter State : Delhi

State not found


        Menu
1. Insert State
2. Find Population
3. Display
4. Exit
choice : 4

Thank You!!
--------------------------------
Process exited after 49.91 seconds with return value 0
Press any key to continue . . .
```