```
//============================================================================
// Name        : Dictionay.cpp
// Author      :
// Version     :
// Copyright   : Your copyright notice
// Description : Hello World in C++, Ansi-style
//============================================================================

#include <iostream>
#include <iomanip>
using namespace std;

class RECORD              //class for person record
{
      string name;
      int roll_no;
      int link;
public:
      RECORD()
    {
         name=" ";
         roll_no= 0;
         link=-1;
    }
      friend class DICTIONARY;
      friend int main();
};

class DICTIONARY              //class for directory
{
    RECORD HT[10];
public:
    void Insert(RECORD P);
    int search(int);
    void display_HT();
    friend int main();
};

int DICTIONARY::search(int s)
{
    int hl,j;
    hl= s%10;
    if(HT[hl].roll_no==s)        //Check home location contains desired record
or not
    {
      return hl;
    }
    else
    {
      j=HT[hl].link;
      while(j!=-1)          //sequentially search in chain
      {
            if(HT[j].roll_no ==s)
                  return j;
            j=HT[j].link;
      }
    }
    return -1;               //Otherwise record not present
}

void DICTIONARY::display_HT()
```

```
{
     cout<<"\n                      Roll Call List                ";
    cout<<"\n-----------------------------------------------------";
    cout<<"\n| Location |    Roll No.  |   Name        |  Link |";
    cout<<"\n-----------------------------------------------------";
    for(int i=0;i<10;i++)
    {
     if(HT[i].roll_no == 0)
           cout<<"\n|    "<<setw(2)<<i<<"    |      --      |      --       |
"<<setw(2)<<HT[i].link<<"  |";
     else
           cout<<"\n|    "<<setw(2)<<i<<"    |
"<<setw(10)<<right<<HT[i].roll_no<<"   | "<<setw(10)<<left<<HT[i].name<<"    |
"<<setw(2)<<HT[i].link<<"  |";
     cout<<"\n-----------------------------------------------------";
    }
}



void DICTIONARY::Insert(RECORD p)
{
    int hl,j,k,i;
    hl = p.roll_no%10;
    if(HT[hl].roll_no== 0)           //Hashed location is empty.
    {
     HT[hl]=p;
    }
    else
    {
     k=(HT[hl].roll_no)%10;
     if(hl==k)                 //Hashed location contains synonym.
     {
            while(HT[k].link!=-1)  // go to end of chain just like link list
                 k=HT[k].link;

            for(i=1;i<10;i++)
            {
               j=(hl+i)%10;
               if(HT[j].roll_no== 0)   // store the new record
               {
                    HT[j]=p;
                    HT[k].link=j;    // Update the link field.
                    break;
               }
            }
            if(i==10)
                 cout<<"\n DICTIONARY is full";

     }
     else                           //Hashed location contains other than
synonym
     {
            RECORD t;
            t=HT[hl];

            while(HT[k].link!=hl)     //Locate the pred. of collided record of
chain
                 k=HT[k].link;
            for(i=1;i<10;i++)         //find the empty location for collided 
record
            {
                j=(hl+i)%10;
                if(HT[j].roll_no== 0)
                   break;
            }
            if(i==10)
                cout<<"\n Table is full";
            else
            {
                if(HT[hl].link!=-1)            //If collided record has 
successors
                {
                     HT[k].link=HT[hl].link;   //Make pred point to succ of 
collided record
                     while(HT[k].link!=-1)      //go to end of chain
                          k=HT[k].link;
                }
                     HT[k].link=j;             //update link field of last 
record with new loc
                     HT[j]=t;                  //put the collided record in new 
loc
                     HT[j].link=-1;           //make the collided record as 
last of chain
                     HT[hl]=p;                //keep the new record at its 
hashed location
            }
      }
    }
}


int main()
{
      DICTIONARY d;
      int ch,rn;
      char c;
      string s;
      do
      {
            cout<<"\n ------------- MENU -------------";
            cout<<"\n 1. INSERT RECORD                 ";
            cout<<"\n 2. SEARCH RECORD                 ";
            cout<<"\n 3. DISPLAY DICTIONARY            ";
            cout<<"\n 4. Exit                          ";
            cout<<"\n ------------------------------------";
            cout<<"\n Enter your choice=>";
            cin>>ch;
            switch(ch)
            {
               case 1:
                     do
                     {
                          RECORD P;
                   cout<<"\n Enter the roll no to insert=>";
                   cin>>P.roll_no;
                   if(d.search(P.roll_no)==-1)
                   {
                   cout<<"\n Enter the name =>";
                   cin>>P.name;
```

```
                  d.Insert(P);
                  }
                  else
                  {
                      cout<<"\n Record already present";
                  }
                  cout<<"\n Do you want to insert anymore record(y/n)=";
                  cin>>c;
                   }while(c=='y'||c=='Y');
                   break;
              case 2:
                      cout<<"\n Enter the roll no to search=>";
                      cin>>rn;
                  if((ch=d.search(rn))==-1)
                  {
                  cout<<"\n Name not present in DICTIONARY";
                  }
                  else
                  {
                  cout<<"\n-------------------------------";
                  cout<<"\n|    Roll No     |  Name  |";
                  cout<<"\n-------------------------------";
                  cout<<"\n|  "<<setw(10)<<left<<d.HT[ch].roll_no<<"  |
"<<setw(10)<<right<<d.HT[ch].name<<"  |";
                  cout<<"\n-------------------------------";
                  }
                   break;
              case 3:
                  d.display_HT();
                   break;
              case 4:
                   cout<<"\n Exiting..........";
                   break;
              default:
                      cout<<"\n Enter the correct choice......!";
           }
      }while(ch!=5);
      return 0;
}
```