Name : Resham Landge
Roll No : 2339

*******************************Assignment No : 9***********************************

Title : Store data of students with telephone no and name in the structure using hashing function for telephone number and implement chaining with and without replacement.

****

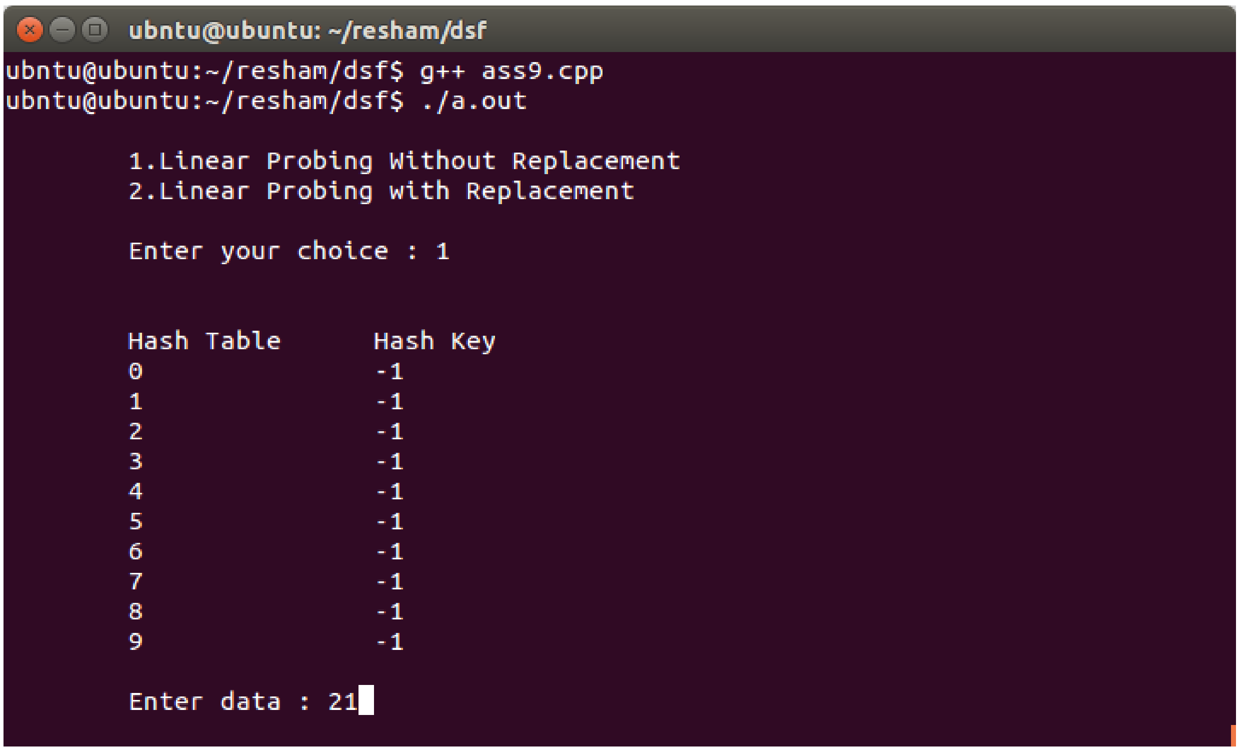
```
#include<iostream>
#define MAX 10
using namespace std;

class Hash
{
        public:
                int table[MAX];

                void linear_without_rep();
                void linear_with_rep();
                int hash(int key);
                int empty(int table[MAX],int loc);
};


int Hash::hash(int key)                         //hash function for creating hash table
{
        return(key % 10);
}

int Hash::empty(int table[MAX],int loc)         //function to find empty fields in table
{
        int i=loc;

        do
        {
                i++;
                i=i % MAX;
        }while(table[i]!=-1 && 1!=loc);         //continue the loop until i & loc is not same

        return i;
}
```

```
void Hash::linear_without_rep()                         //linear probing without replacement
{
        int key,loc,pos,i=0;
        char ch;

        for(i=0; i<MAX; i++)
                table[i]=-1;
        cout<<"\n\n\tHash Table\tHash Key";
        for(i=0; i<MAX; i++)
                cout<<"\n\t"<<i<<"\t\t"<<table[i];

        i=0;

        do
        {
                cout<<"\n\n\tEnter data : ";
                cin>>key;

                loc=hash(key);                          //call hash key function to find location

                if(table[loc]==-1)                      //if loc is empty then copy to it
                        table[loc]=key;
                else
                {
                        pos=empty(table,loc);           //if loc not empty then

                        if(pos!=loc)                    //find its next empty space in table
                                table[pos]=key;
                        else
                                cout<<"\n\tHash Table Full ";
                }

                cout<<"\n\tHash Table\tHash key";
                for(i=0; i<MAX; i++)
                        cout<<"\n\t"<<i<<"\t\t"<<table[i];

                cout<<"\n\n\tDo U want more element(y/n) : ";
                cin>>ch;
        }while(ch=='Y' || ch=='y');
}

void Hash::linear_with_rep()                            //linear probing with replacement
{
        int key,i=0,loc,pos;
```

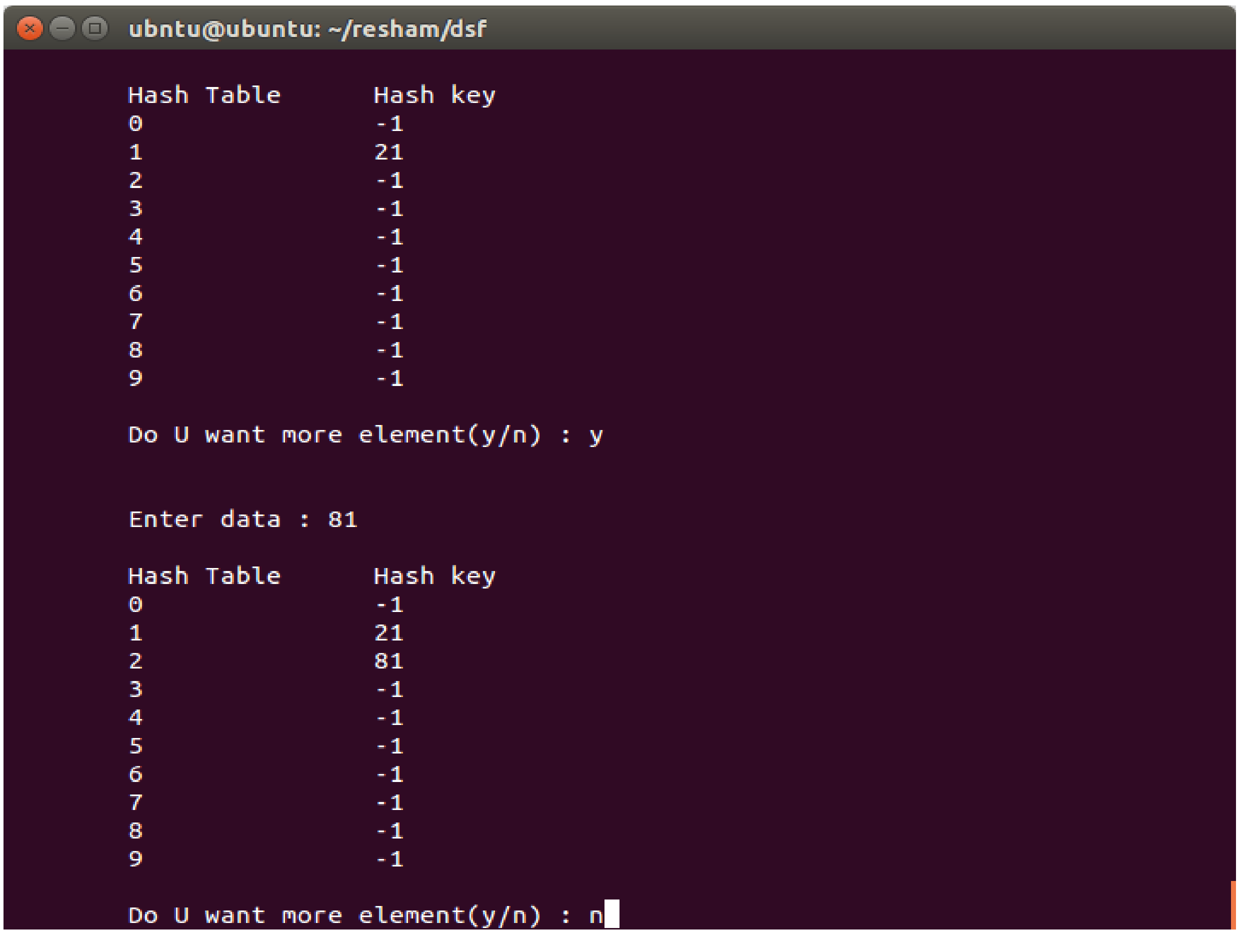
```
char ch;

for(i=0; i<MAX; i++)
        table[i]=-1;

cout<<"\n\n\tHash Table\tHash Key";
for(i=0; i<MAX; i++)
        cout<<"\n\t"<<i<<"\t\t"<<table[i];

i=0;

do
{
        cout<<"\n\tEnter data : ";
        cin>>key;

        loc=hash(key);                          //call hash key function to find location

        if(table[loc]==-1)                      //if location is empty then copy it
                table[loc]=key;
        else
        {
                pos=empty(table,loc);           //if it contains data then it is null

                if(pos==loc)
                        cout<<"\n\tHash table Full";
                else
                {
                        if(loc==hash(table[loc]))       //replace that key
                                table[pos]=key;
                        else
                        {
                                table[pos]=table[loc];
                                table[loc]=key;
                        }
                }
        }

        cout<<"\n\tHash table\tHash Key";

        for(i=0; i<MAX; i++)
                cout<<"\n\t"<<i<<"\t\t"<<table[i];

        cout<<"\n\n\tDo U want more element(y/n) : ";
```

```
                cin>>ch;
        }while(ch=='Y' || ch=='y');
}

int main()
{
        Hash h;
        int choice;
        char ch;

        do
        {
                cout<<"\n\t1.Linear Probing Without Replacement\n\t2.Linear Probing with
                Replacement\n";
                cout<<"\n\tEnter your choice : ";
                cin>>choice;

                switch(choice)
                {
                        case 1:
                                h.linear_without_rep();
                                break;

                        case 2:
                                h.linear_with_rep();
                                break;
                }
                cout<<"\n\tDo U want to continue(y/n) : ";
                cin>>ch;
        }while(ch=='Y'||ch=='y');

        return 1;
}
```

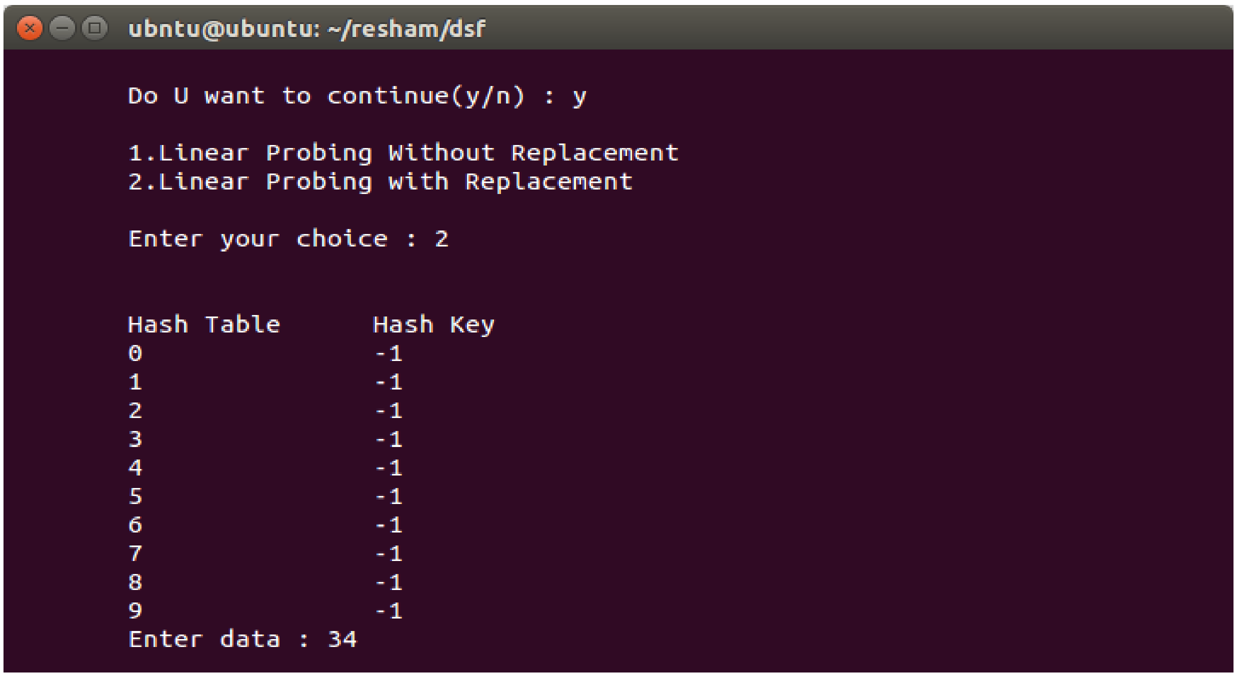
Output :



```
ubntu@ubuntu:~/resham/dsf$ g++ ass9.cpp
ubntu@ubuntu:~/resham/dsf$ ./a.out

        1.Linear Probing Without Replacement
        2.Linear Probing with Replacement

        Enter your choice : 1


        Hash Table      Hash Key
        0               -1
        1               -1
        2               -1
        3               -1
        4               -1
        5               -1
        6               -1
        7               -1
        8               -1
        9               -1

        Enter data : 21
```

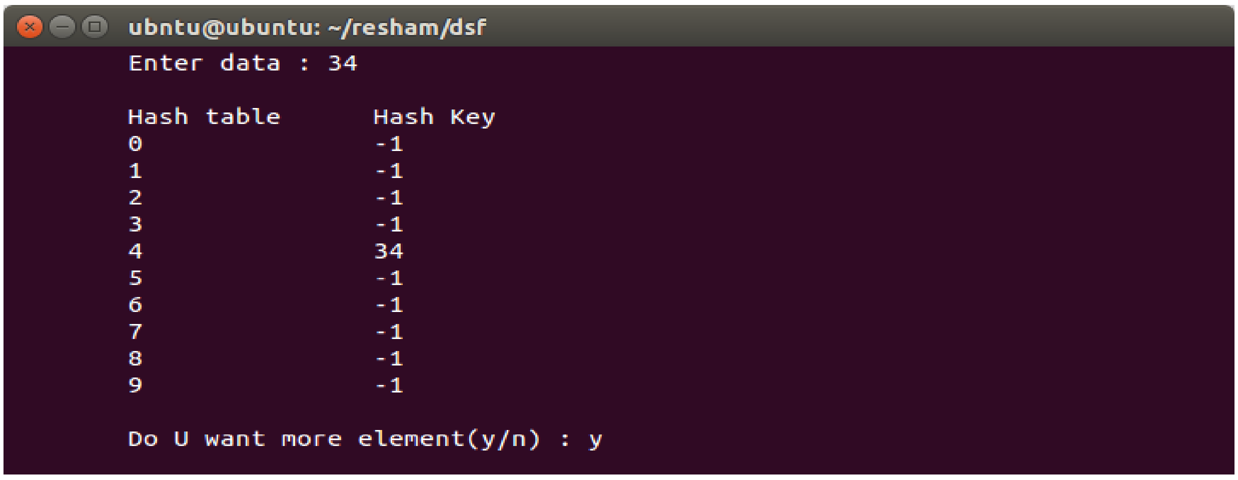


```
        Hash Table      Hash key
        0               -1
        1               21
        2               -1
        3               -1
        4               -1
        5               -1
        6               -1
        7               -1
        8               -1
        9               -1

        Do U want more element(y/n) : y


        Enter data : 81

        Hash Table      Hash key
        0               -1
        1               21
        2               81
        3               -1
        4               -1
        5               -1
        6               -1
        7               -1
        8               -1
        9               -1

        Do U want more element(y/n) : n
```



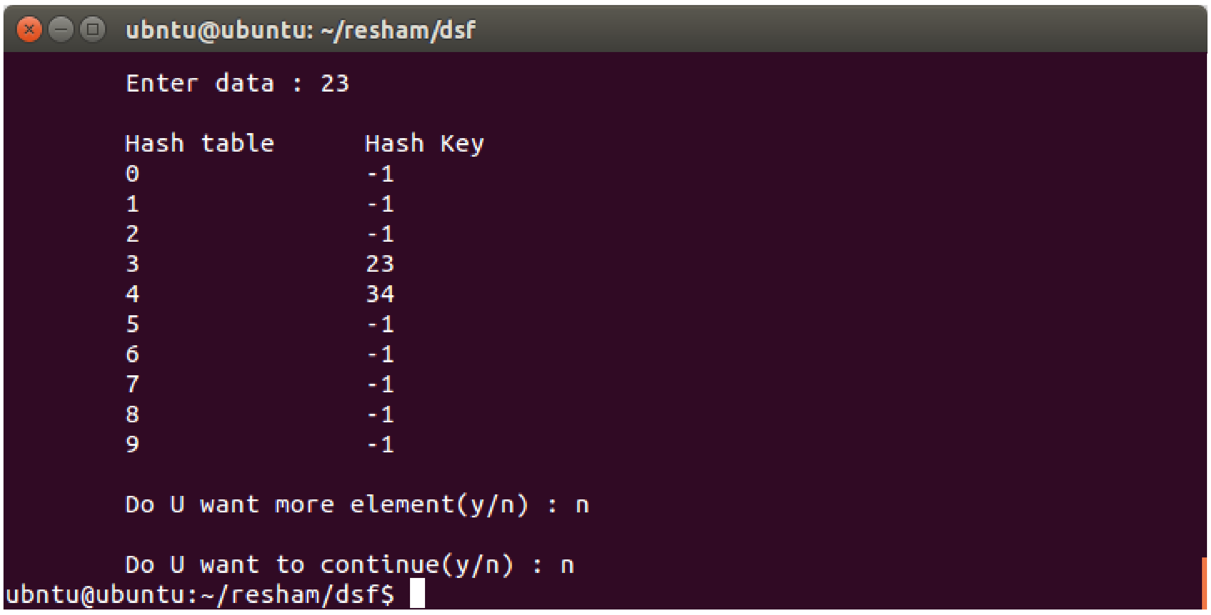
ubntu@ubuntu: ~/resham/dsf

```
        Do U want to continue(y/n) : y

        1.Linear Probing Without Replacement
        2.Linear Probing with Replacement

        Enter your choice : 2


        Hash Table      Hash Key
        0               -1
        1               -1
        2               -1
        3               -1
        4               -1
        5               -1
        6               -1
        7               -1
        8               -1
        9               -1
        Enter data : 34
```



ubntu@ubuntu: ~/resham/dsf

```
        Enter data : 34

        Hash table      Hash Key
        0               -1
        1               -1
        2               -1
        3               -1
        4               34
        5               -1
        6               -1
        7               -1
        8               -1
        9               -1

        Do U want more element(y/n) : y
```



ubntu@ubuntu: ~/resham/dsf

```
        Enter data : 23

        Hash table      Hash Key
        0               -1
        1               -1
        2               -1
        3               23
        4               34
        5               -1
        6               -1
        7               -1
        8               -1
        9               -1

        Do U want more element(y/n) : n

        Do U want to continue(y/n) : n
ubntu@ubuntu:~/resham/dsf$
```