Code:

#include<iostream>

using namespace std;

#include<string.h>

#define MAX 10

struct hash

{ char name[20];

long int tele;

int chain;

}rec[MAX];

class hashtable

{ public:hashtable()

{

for(int i=0;i<MAX;i++)

{ rec[i].tele=0;

strcpy(rec[i].name,"-");

rec[i].chain=-1;

}

}

int hashing(long int tel)

{ return (tel%MAX);

}

void with\_repl(struct hash h);

void without\_repl(struct hash h);

void disp();

};

void hashtable::with\_repl(struct hash h)

{ int k=hashing(h.tele);

int p=k;

int i,change\_pos;

if(rec[k].tele==0)

{

rec[k].tele=h.tele;

strcpy(rec[k].name,h.name);

}

else if(hashing(rec[k].tele)==hashing(h.tele))

{

i=k+1;

do

{

if(i>=MAX)

i=i%MAX;

if(hashing(rec[i].tele)==k)

p=k;

if(rec[i].tele==0)

{

rec[i].tele=h.tele;

strcpy(rec[i].name,h.name);

int j=k;

do

{

if(hashing(rec[j].tele)==k && rec[j].chain==-1)

{

rec[j].chain=i;

break;

}

j++;

if(j>=MAX)

j=j%MAX;

}while(j%MAX!=k);

break;

}

i++;

}while(i%MAX!=k);

if(i==k)

cout<<"\nHash table is full.";

}

else

{

hash temp;

temp.tele=rec[k].tele;

strcpy(temp.name,rec[k].name);

temp.chain=rec[k].chain;

rec[k].tele=h.tele;

strcpy(rec[k].name,h.name);

rec[k].chain=-1;

i=k+1;

if(i>=MAX)

i=i%MAX;

do

{

if(rec[i].tele==0)

{

rec[i].tele=temp.tele;

strcpy(rec[i].name,temp.name);

rec[i].chain=temp.chain;

change\_pos=i;

break;

}

i++;

}while(i%MAX!=k);

for(i=0;i<MAX;i++)

{

if(rec[i].chain==k)

{

if(change\_pos>=MAX)

change\_pos=change\_pos-MAX;

rec[i].chain=change\_pos;

break;

}

}

}

}

void hashtable::without\_repl(struct hash h)

{

int k=hashing(h.tele);

int p=k;

int i;

if(rec[k].tele==0)

{

rec[k].tele=h.tele;

strcpy(rec[k].name,h.name);

}

else

{

i=k+1;

do

{

if(i>=MAX)

i=i%MAX;

if(hashing(rec[i].tele)==k)

p=k;

if(rec[i].tele==0)

{

rec[i].tele=h.tele;

strcpy(rec[i].name,h.name);

do

{

if(hashing(rec[p].tele)==k && rec[p].chain==-1)

{

rec[p].chain=i;

break;

}

p++;

p=p%MAX;

}while(p%MAX!=i);

break;

}

i++;

}while(i%MAX!=k);

if(i==k)

cout<<"\nHash table is full.";

}

}

void hashtable::disp()

{

cout<<"\nIndex\tTele. No.\tName\tChain";

int i;

for(i=0;i<MAX;i++)

cout<<"\n"<<i<<"\t"<<rec[i].tele<<"\t\t"<<rec[i].name<<"\t"<<rec[i].chain;

}

int main()

{

hashtable ht;

hash h;

int ch;

while(1)

{ cout<<"\n1.Insert w/o replacement\n2.Insert with replacement\n3.Display\n4.Exit\nEnter option : ";

cin>>ch;

switch(ch)

{

case 1:

cout<<"Enter tel. no : ";

cin>>h.tele;

cout<<"Enter name : ";

cin>>h.name;

ht.without\_repl(h);

break;

case 2:

cout<<"Enter tel. no : ";

cin>>h.tele;

cout<<"Enter name : ";

cin>>h.name;

ht.with\_repl(h);

break;

case 3:

ht.disp();

break;

case 4:

return 0;

}

}

return 0;

}

Output:



