#include <iostream>

#include<stdlib.h>

#include<string.h>

#define MAX 10

using namespace std;

struct hash1

{

int id;

char name[20];

}rec[MAX],tmp;

class hashtable

{

public:

void initialize();

int hashing(int);

void wreplace(struct hash1);

void place\_at\_next\_slot(int, struct hash1);

void display();

};

void hashtable::initialize()

{

int i;

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

{

rec[i].id = -1;

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

}

}

int hashtable::hashing(int no)

{

int key;

key = no%MAX;

return(key);

}

void hashtable::place\_at\_next\_slot(int k, struct hash1 h)

{

int i,z;

for(i =1; i<MAX; i++)

{

z=(k+i)%MAX;

if(rec[z].id == -1)

{

rec[z].id = h.id;

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

break;

}

}

}

void hashtable::display()

{

cout<<"\n Collision Handling :\n\t----------------------------------";

cout<<"\n\t Index ID Name \n\t------------------------------\n";

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

{

cout<<"\t"<<i <<"\t "<<rec[i].id;

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

cout<<"\n";

}

cout<<"\n\t--------------------------------\n";

}

void hashtable::wreplace(struct hash1 h)

{

int k,p,change\_pos;

k = hashing(h.id);

if(rec[k].id == -1)

{

rec[k].id=h.id;

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

}

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

place\_at\_next\_slot(k,h);

else{

tmp.id = rec[k].id;

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

rec[k].id = h.id;

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

place\_at\_next\_slot(k,tmp);

}

}

int main()

{

hashtable h;

int ch,ch1,key;

char ans;

struct hash1 temph;

do

{

cout<<"\n Collision Handling ";

cout<<"\n1.With Replacement ";

cout<<"\n 2. Display Record\n 3. Exit \n Enter your choice : ";

cin>>ch;

switch(ch)

{

case 1:

h.initialize();

do{

cout<<"\n Enter ID: ";

cin>>temph.id;

cout<<"\n Enter Name: ";

cin>>temph.name;

h.wreplace(temph);

cout<<"\n Do you want to add more elements(0/1)?";

cin>>ch1;

}while(ch1!=0);

break;

case 2:

h.display();

break;

case 3:

exit(0);

}

}while (ch!=5);

return 0;

}