#include<stdio.h>

#define size 10

int count=0;

struct hashtable

{

int data;

int delstatus;

};

struct hashtable ht[size];

void initHT()

{

int i;

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

{

ht[i].delstatus = 1;

}

}

int isFull()

{

if(count==size)

return 1;

else

return 0;

}

int isEmpty()

{

if(count == 0)

return 1;

else

return 0;

}

int addData()

{

int data, key, i;

if(!isFull())

{

printf("\nEnter data: ");

scanf("%d", &data);

key = data % size;

if(ht[key].delstatus == 1)

{

ht[key].data =data;

ht[key].delstatus=0;

printf("\nData added to table");

count++;

}

else

{

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

{

key=(key+1)%size;

if(ht[key].delstatus==1)

{

ht[key].data=data;

ht[key].delstatus=0;

printf("\nData is added to table");

count++;

break;

}

}

}

}

else

{

printf("\nHash Table is full");

}

}

int delData()

{

int data, key, i, flag=0;

if(!isEmpty())

{

printf("\nEnter data to delete: ");

scanf("%d",&data);

key=data%size;

if(ht[key].delstatus==0 && ht[key].data==data)

{

ht[key].delstatus=1;

printf("\nData deleted from table");

count--;

}

else

{

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

{

key=(key+1)%size;

if(ht[key].data==data)

{

ht[key].delstatus=1;

printf("\nData deleted from table");

count--;

flag=i;

break;

}

}

if(flag==0)

printf("\nData to be deleted not found in table");

}

}

else

printf("\nHash Table is empty");

}

int display()

{

int i;

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

{

if(ht[i].delstatus==0)

{

printf("|%d|", ht[i].data);

}

else

printf("| |");

}

}

int main()

{

int ch;

initHT();

do

{

printf("\nMain Menu: ");

printf("\n1)Add data\n2)Delete data\n3)Dispaly data\n4)Exit\nEnter a choice: ");

scanf("%d", &ch);

switch(ch)

{

case 1 : addData();

break;

case 2 : delData();

break;

case 3 : display();

break;

case 4 : printf("\nExit from program");

break;

default : printf("\nWrong choice");

}

}while(ch!=4);

}