/\*Design and develop a program in C that uses hash function H:k->L as H(K)=K mod m ( reminder method)

and implement hashing technique to map a given key K to the address space L. Resolve the collission (if any)

using linear probing. \*/

#include <stdio.h>

#include <stdlib.h>

#define MAX 10

int create(int num)

{

int key;

key=num%100;

return key;

}

void linear\_prob(int a[MAX], int key, int num)

{

int flag, i, count=0;

flag=0;

if(a[key]== -1)

{

a[key] = num;

}

else

{

printf("\nCollision Detected...!!!\n");

i=0;

while(i<MAX)

{

if (a[i]!=-1)

count++;

i++;

}

printf("Collision avoided successfully using LINEAR PROBING\n");

if(count == MAX)

{

printf("\n Hash table is full");

display(a);

exit(1);

}

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

if(a[i] == -1)

{

a[i] = num;

flag =1;

break;

}

i=0;

while((i<key) && (flag==0))

{

if(a[i] == -1)

{

a[i] = num;

flag=1;

break;

}

i++;

}

}

}

void display(int a[MAX])

{

int i;

printf("\n\n Displaying all: The hash table is\n");

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

printf("\n %d %d ", i, a[i]);

printf("\n Filtered Display: The hash table is\n\n");

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

if(a[i]!=-1)

{

printf("\n %d %d ", i, a[i]);

continue;

}

}

void main()

{

int a[MAX],num,key,i;

int ans=1;

printf(" Hashing : Collision handling by linear probing : \n");

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

{

a[i] = -1;

}

do

{

printf("\n Enter the data");

scanf("%4d", &num);

key=create(num);

linear\_prob(a,key,num);

printf("\n Do you wish to continue ? (1/0) ");

scanf("%d",&ans);

}while(ans);

display(a);

}