//WAP to implement Hash Table

#include <stdio.h>

#include <stdlib.h>

int ht[10], i, found = 0, key;

void insert();

void search();

void delete();

void display();

int main()

{

int option;

for (i = 0; i < 10; i++) // to initialize every element as ‘–1’

ht[i] = -1;

do {

printf("\n\*\*\*List Of Operations\*\*\*\n1. Insert \n2. Search \n3. Delete \n4. Display \n5. Exit");

printf("\nEnter your option: ");

scanf("%d", &option);

switch (option) {

case 1:

insert();

break;

case 2:

search();

break;

case 3:

delete();

break;

case 4:

display();

break;

case 5:

printf("\n\tEXIT POINT!");

}

} while (option != 5);

return 0;

}

void insert()

{

int val, f = 0;

printf("Enter the element to be inserted: ");

scanf("%d", &val);

key = (val % 10) - 1;

if (ht[key] == -1)

{

ht[key] = val;

}

else

{

if (key < 9) {

for (i = key + 1; i < 10; i++) {

if (ht[i] == -1) {

ht[i] = val;

break;

}

}

}

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

{

if (ht[i] == -1) {

ht[i] = val;

break;

}

}

}

}

void search()

{

int val, flag = 0;

printf("Enter the element to be searched: ");

scanf("%d", &val);

key = (val % 10) - 1;

if (ht[key] == val)

flag = 1;

else

{

for (i = key + 1; i < 10; i++) {

if (ht[i] == val) {

flag = 1;

key = i;

break;

}

}

}

if (flag == 0)

{

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

{

if (ht[i] == val)

{

flag = 1;

key = i;

break;

}

}

}

if (flag == 1)

{

found = 1;

printf("\nThe item searched was found at position %d\n", key + 1);

}

else

{

key = -1;

printf("\nThe item searched was not found in the hash table\n");

}

}

void delete()

{

search();

if (found == 1)

{

if (key != -1)

{

printf("\nThe element deleted is %d\n", ht[key]);

ht[key] = -1;

}

}

}

void display()

{

for (i = 0; i < 10; i++)

printf("\t%d", ht[i]);

}







