S.No: 21	Exp. Name: ***Collision resolution techniques : linear probing***	Date:2024-06-02

Aim:

Write a C program to implement Linear Probing.

Source Code:

HashingMain2.c

```
#include <stdio.h>
#include <stdlib.h>
#include "HashingLinearProbing.c"
int main() {
   int x, op, i = 0;
   for (i = 0; i < SIZE; i++)
      HashTable[i] = -1;
   while (1) {
      printf("1.Insert 2.Delete 3.Search 4.Print 5.Exit\n");
      printf("Enter your option : ");
      scanf("%d", &op);
      switch (op) {
         case 1:  printf("Enter an element to be inserted : ");
               scanf("%d", &x);
               insert(x);
               break;
         case 2:
               printf("Enter an element to be deleted : ");
               scanf("%d", &x);
               deleteElement(x);
               break;
         case 3:
               printf("Enter an element to be searched : ");
               scanf("%d", &x);
               search(x);
               break;
         case 4:
               print();
               break;
         case 5:  exit(0);
      }
   }
}
```

HashingLinearProbing.c

```
#define SIZE 10
int HashTable[SIZE];

int hash(int x) {
    return x % SIZE;
}

void insert(int x) {
    int index, start;
```


ID: 92132321021

```
    index = hash(x);
    start = index;
    while (HashTable[index] != -1) {
        if (HashTable[index] == -1) {
            break;
        }
        index = (index + 1) % SIZE;
        if (index == start) {
            printf("Hash table is full. So cannot insert the element.\n");
            return;
        }
    }
    HashTable[index] = x;
    printf("Successfully inserted.\n");
}

void deleteElement(int x) {
    int index, start;
    index = hash(x);
    start = index;
    while (HashTable[index] != x) {
        if (HashTable[index] == x) {
            break;
        }
        index = (index + 1) % SIZE;
        if (index == start) {
            printf("Element not found. So cannot delete the element.\n");
            return;
        }
    }
    HashTable[index] = -1;
    printf("Successfully deleted.\n");
}

void search(int x) {
    int index, start;
    index = hash(x);
    start = index;
    while (HashTable[index] != x) {
        if (HashTable[index] == x) {
            break;
        }
        index = (index + 1) % SIZE;
        if (index == start) {
            printf("Element not found.\n");
            return;
        }
    }
    printf("Element found.\n");
}

void print() {
    int i;
    for (i = 0; i < SIZE; i++) {
        if (HashTable[i] != -1) {
            printf("[%d]=>%d ", i, HashTable[i]);
```

```
        }
    }
    printf("\n");
}
```


Execution Results - All test cases have succeeded!

Test Case - 1
User Output
1.Insert 2.Delete 3.Search 4.Print 5.Exit 1
Enter your option : 1
Enter an element to be inserted : 11
Successfully inserted. 1
1.Insert 2.Delete 3.Search 4.Print 5.Exit 1
Enter your option : 1
Enter an element to be inserted : 22
Successfully inserted. 1
1.Insert 2.Delete 3.Search 4.Print 5.Exit 1
Enter your option : 1
Enter an element to be inserted : 33
Successfully inserted. 1
1.Insert 2.Delete 3.Search 4.Print 5.Exit 1
Enter your option : 1
Enter an element to be inserted : 43
Successfully inserted. 1
1.Insert 2.Delete 3.Search 4.Print 5.Exit 1
Enter your option : 1
Enter an element to be inserted : 53
Successfully inserted. 4
1.Insert 2.Delete 3.Search 4.Print 5.Exit 4
Enter your option : 4
[1]=>11 [2]=>22 [3]=>33 [4]=>43 [5]=>53 1
1.Insert 2.Delete 3.Search 4.Print 5.Exit 1
Enter your option : 1
Enter an element to be inserted : 44
Successfully inserted. 4
1.Insert 2.Delete 3.Search 4.Print 5.Exit 4
Enter your option : 4
[1]=>11 [2]=>22 [3]=>33 [4]=>43 [5]=>53 [6]=>44 3
1.Insert 2.Delete 3.Search 4.Print 5.Exit 3
Enter your option : 3
Enter an element to be searched : 34
Element not found. 2
1.Insert 2.Delete 3.Search 4.Print 5.Exit 2
Enter your option : 2
Enter an element to be deleted : 33
Successfully deleted. 4
1.Insert 2.Delete 3.Search 4.Print 5.Exit 4
Enter your option : 4
[1]=>11 [2]=>22 [4]=>43 [5]=>53 [6]=>44 5

```
1.Insert 2.Delete 3.Search 4.Print 5.Exit 5
Enter your option :  5
```