**Assignment 12**

**Name** : Komal Potdar

**Roll No**.: 92

**PRN No**.: 12320165

**Div**: CS B SY

**Batch**: 3

1. **Implement following collison handling techniques for Hash Table :**
2. **Linear Probing**
3. **Double Hashing**
4. **Quadratic Hashing**

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #define SIZE 10  int hash(int key) {  return key % SIZE;  }  int linearProbe(int H[], int key) {  int index = hash(key);  int i = 0;  while (H[(index + i) % SIZE] != 0)  i++;  return (index + i) % SIZE;  }  int secondaryHash(int key) {  return 7 - (key % 7);  }  int doubleHashing(int H[], int key) {  int index = hash(key);  int step = secondaryHash(key);  int i = 0;  while (H[(index + i \* step) % SIZE] != 0) {    i++;  }  return (index + i \* step) % SIZE;  }  int quadraticProbe(int H[], int key) {  int index = hash(key);  int i = 0;  while (H[(index + i \* i) % SIZE] != 0)  i++;  return (index + i \* i) % SIZE;  }  void insertLinearProbe(int H[]) {  int key;  char choice;  do {  printf("Enter key to insert: ");  scanf("%d", &key);  int index = hash(key);  if (H[index] != 0)  index = linearProbe(H, key);  H[index] = key;  printf("Do you want to insert another key? (y/n): ");  scanf(" %c", &choice);  } while (choice == 'y' || choice == 'Y');  }  void insertDoubleHashing(int H[]) {  int key;  char choice;  do {  printf("Enter key to insert: ");  scanf("%d", &key);  int index = hash(key);  int result = doubleHashing(H, key);  if (result != -1) {  index = result;  } else {  printf("Hash table is full. Cannot insert %d.\n", key);  continue; // Move to the next iteration  }  H[index] = key;  printf("Do you want to insert another key? (y/n): ");  scanf(" %c", &choice);  } while (choice == 'y' || choice == 'Y');  }  void insertQuadraticProbe(int H[]) {  int key;  char choice;  do {  printf("Enter key to insert: ");  scanf("%d", &key);  int index = hash(key);  if (H[index] != 0)  index = quadraticProbe(H, key);  H[index] = key;  printf("Do you want to insert another key? (y/n): ");  scanf(" %c", &choice);  } while (choice == 'y' || choice == 'Y');  }  void display(int H[]) {  for (int i = 0; i < SIZE; i++)  printf("%d ", H[i]);  printf("\n");  }  int main() {  int HT\_linear[SIZE] = {0};  int HT\_double[SIZE] = {0};  int HT\_quadratic[SIZE] = {0};  int choice;  do {  printf("\nCollision Handling Techniques:\n");  printf("1. Linear Probing\n");  printf("2. Double Hashing\n");  printf("3. Quadratic Probing\n");  printf("4. Display Hash Tables\n");  printf("5. Exit\n");  printf("Enter your choice: ");  scanf("%d", &choice);  switch (choice) {  case 1:  insertLinearProbe(HT\_linear);  break;  case 2:  insertDoubleHashing(HT\_double);  break;  case 3:  insertQuadraticProbe(HT\_quadratic);  break;  case 4:  printf("\nLinear Probing: ");  display(HT\_linear);  printf("Double Hashing: ");  display(HT\_double);  printf("Quadratic Probing: ");  display(HT\_quadratic);  break;  case 5:  printf("Exiting...\n");  break;  default:  printf("Invalid choice!\n");  }  } while (choice != 5);  return 0;  } |

**Output:**







