21/2/24

LAB - 10

```
#include <stdio.h>
#include <stdlib.h>
#define MAX_MEMORY_LOCATIONS 10
int hashTable[MAX_MEMORY_LOCATIONS] = {0};
int status[MAX_MEMORY_LOCATIONS] = {0};

void initializeHashTable(int size){
    for(int i=0; i<size; i++){
        status[i] = 0;
    }
}

int hashFunction(int key, int size){
    return key % size;
}

int linearProbe(int hashValue, int attempt,
                int size){
    return (hashValue + attempt) % size;
}

void insertEmployee(int size, int key){
    int hashValue = hashFunction(key, size);
    int index = hashValue;
    int attempt = 1;
    while(status[index] == 1){
        index = linearProbe(hashValue, attempt,
                size);
        attempt++;
    }
    hashTable[index] = key;
```

```
    status[index] = 1;
}

void displayHashTable(int size){
    printf("\n Hash Table:\n");
    printf(" Index\t key\n");

    for(int i=0; i<size; i++){
        printf("%d\t", i);
        if(status[i]==1){
            printf("%d\n", hashTable[i]);
        }else{
            printf(" Empty\n");
        }
    }
}

int main(){
    int m = MAX_MEMORY_LOCATIONS;
    initializeHashTable(m);
    int n;
    printf("Enter the number of employee
            records:");
    scanf("%d", &n);
    int keys[n];

    printf("Enter the employee keys:\n");
    for(int i=0; i<n; i++){
        printf("Employee %d:", i+1);
        scanf("%d", &keys[i]);
    }
    for(int i=0; i<n; i++){
```

```
        insertEmployee(m, keys[i]);
    }

    displayHashTable(m);

    return 0;
}
```

OUTPUT -

```
Enter the number of employee records: 6
Enter the employee keys:
Employee 1 : 85
Employee 2 : 56
Employee 3 : 23
Employee 4 : 35
Employee 5 : 43
Employee 6 : 11

Hash Table:
Index    Key
0        Empty
1        11
2        Empty
3        23
4        43
5        85
6        56
7        35
8        Empty
9        Empty
```