Experiment Name / No.: 18

HASHING

AIM

write a menu driven (program to implement hash table and the following collision resolution techniques - i) linear probing ii) quadratic probing iii) chaining

ALGORITHM

```
1 Start
2 set all elements in hash tables = -1
3 Enter hash table size.
4 choose between collision resolution method
5 case 1 : linear probing
   5.1 Enter key
   5.2 set finalhash = key % size
   5.3 if lin-prop-array(hash) != -1, collision occured
        5.3.1 repeat until i<size, lin-prob-array(finalhash) = -1
              5.3.1.1 set final hash = (hash + 1) % size.
   5.4 set lin-prob-array[finalhash) = key
   5.5 set quad-prob-array (finalhash) = key
6 case 2: quadratic probing
   6.1 Enter key
   6.2 Enter values for c1 & c2
   6.3 set finalhash = key % size
   6.4 if quad-prob-array(hash) != -1, collision occured.
```

```
        6.4.1 Repeat until i<size, quad-prob rrray (finalhash) != -1
              6.4.1.1 set finalhash = (hash + c1i + c2i^2) % size
    6.5 set quad-prob-array (final hash) = key
7   case 3 : chaining
    7.1 Enter key
    7.2 set hash = key % size.
    7.3 create newnode with value as key
    7.4    if chaining-array(hash) = NULL
           7.5 set chaining-arras (hash) = new node.
    7.5 else
        7.5.1 set ptr = chaining-array(hash), traverse
              to the end of linked list
        7.5.2 Add newnode to end of list
8   stop
```

CONCLUSION

The program has been executed correctly and output has been verified