Hashing

Task

Implement the following Collision handling techniques:

1. Linear Probing
2. Quadratic Probing
3. Double Hashing

The first input line should be (choice, N, Q, S), where ‘choice’ can be 1/2/3 corresponding to linear/quadratic/double hashing. N represents the size of the HashTable. Q represents the number of entries, and S represents the number of searches.
Then, there will be Q numbers given as input, followed by S numbers to find from the hashtable.

The primary hash function should be Hash(x) = (x % TableSize)

Choose a suitable hash function for double hashing in addition to the function as mentioned above (as it needs two)

Sample Input	**Sample Output**
1 10 8 3 35 45 73 36 5 24 13	 Inserted : Index-5 (L.F=.1) Collision : Index-5 Inserted : Index-6 (L.F=.2) Inserted : Index-3 (L.F=.3) Collision : Index-6 Inserted : Index-**7** (L.F=.4) Collision: Index-5 Collision: Index-6 **Collision: Index-7** Inserted : Index-**8** (L.F=.5) Inserted : Index-4 (L.F=.6) Collision: Index-3 Collision: Index-4 Collision: Index-5 Collision: Index-6 Collision: Index-7 **Collision: Index-8** **Input Abandoned**

99 **24** **13** **5**	**Inserted : Index-9 (L.F=.7)** **Found : Index-4** **Not Found!** **Found : Index-8**
2 8 7 4 67 15 86 63 47 33 8 **63** **34** **8** **47**	 Inserted : Index-3 (L.F = 0.125) Inserted : Index-7 (L.F = 0.25) Inserted : Index-6 (L.F = 0.375) Collision: Index-7 Inserted : Index-0 (L.F = 0.5) Collision: Index-7 Collision: Index-0 Collision: Index-3 Collision: Index-0 Collision: Index-7 Collision: Index-0 Input Abandoned Inserted : Index-1 (L.F = 0.625) Collision: Index-0 Collision: Index-1 Inserted : Index-4 (L.F = 0.75) **Found : Index-0** **Not Found!** **Found : Index-4** **Not Found!**

Note:

- L.F means Load Factor.
- If you fail to insert the number within six attempts, abandon that number.