Assignment 4

Report

Created two template hash table classes. First one utilizes a dynamic array that is created and initialized in the constructer for keys and objects. Insert function that uses linear probing which finds the next empty spot to place the object in the table.. Remove function works by looking for the key in the table and turning the spot back to empty. Print function displays each key and its corresponding object. Collision Rate function Displays the number of collisions that happened and the collision rate.

Similarly, the separate chaining hash table uses nodes to store the data and the key. Insert function takes in a key and the object then it places the object in the corresponding index if the index is not empty the object is placed as the next* on that index. Remove function takes in a key and checks the table for the index holding that key finds the corresponding object pointer and deletes it. Print functions showcases each key and its object as well as the way they are stored. Collision Rate function Displays the number of collisions that happened and the collision rate.

The has function I use is a simplified implementation of the murmur3 algorithm. The hash function takes in the key and in a for loop iterates each character in the key. An XOR operation is performed on the character and unsigned int h. h is multiplied by a constant number that makes it more random and shifted to the right by 15 bits. The result is finally modulus with the size of the hash table to return the index. Using this helped lower the collision rate from 5 to 3 a decently big improvement compared to the standard hash function. Although it had less noticeable improvements in the dynamic array hash table.

The separate chaining hash table was easier to implement and had fewer collisions than the dynamic array one. However, from research the amount of memory it takes up might make it impractical for huge hash tables.

ChainHashTable:

```
Choose between linear probing and chaining
1- Linear Probing
2- Chaining
2
Index 0: Key Mariam:Mariam 32 8000 2 -> Key Aya:Aya 26 6000 3 ->
Index 1: Key Fawzy:Fawzy 45 5000 8 -> Key Yara:Yara 19 2000 0 ->
Index 2:
Index 3: Key Ayman:Ayman 33 4000 8 -> Key Abdallah:Abdallah 29 7000 4 ->
Index 4: Key Mina:Mina 30 10000 4 ->
Index 5:
Index 6:
Index 7: Key Roshdy:Roshdy 28 9000 3 ->
Index 8: Key Fatma:Fatma 21 3000 1 ->
Index 9:
Collision rate: 0.333333
Number of collisions: 3

C:\Users\M.H\Desktop\University\Assignments\Assignment-4\x64\Debug\Assignment-4.exe (process 28984) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

LinearHashTable:

```
Choose between linear probing and chaining
1- Linear Probing
2- Chaining
1
0: Mariam Mariam 32 8000
1: Aya Aya 26 6000
2: Yara Yara 19 2000
3: Ayman Ayman 33 4000
4: Abdallah Abdallah 29 7000
7: Roshdy Roshdy 28 9000
8: Fatma Fatma 21 3000
Collision Rate: 0.6
Number of Collisions: 6

C:\Users\M.H\Desktop\University\Assignments\Assignment-4\x64\Debug\Assignment-4.exe (process 31584) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
```