

The screenshot of the output of the **chain** hash table. All the functions are called in the main to be tested as shown in the screenshot

```
Microsoft Visual Studio Debug Console
This is the table using chaining
0 :
1 :
2 : Ayman --->
3 : Roshdy ---> Aya --->
4 :
5 :
6 :
7 : Yara ---> Abdallah --->
8 :
9 : Mina ---> Fawzy ---> Mariam ---> Fatma --->

Rate of collision is
55.5556%
-----
table after calling the remove function to Yara
0 :
1 :
2 : Ayman --->
3 : Roshdy ---> Aya --->
4 :
5 :
6 :
7 : Abdallah --->
8 :
9 : Mina ---> Fawzy ---> Mariam ---> Fatma --->
-----
```

The screenshot of the output of the **Linear** hash table. All the functions are called in the main to be tested as shown in the screenshot

```
This is the table using linear probing
0 : Fawzy
1 : Mariam
2 : Ayman
3 : Roshdy
4 : Aya
5 : Fatma
6 :
7 : Yara
8 : Abdallah
9 : Mina

Rate of collision is
55.5556%
-----
table after calling the remove function to Yara
0 : Fawzy
1 : Mariam
2 : Ayman
3 : Roshdy
4 : Aya
5 : Fatma
6 :
7 :
8 : Abdallah
9 : Mina

A:\Development\C++ - CS 2\Assignment 4\x64\Debug\Assignment 4.exe (process 35436) exited with code 0.
Press any key to close this window . . .
```

As shown, the two collision rates are the same because I chose the same hash algorithm for both classes.

I think the chain method is better because, as seen in the screenshots, it has an unlimited number of potential nodes to be added; however, the linear method can only bear one more node in the table and then it is full and cannot be used to insert more nodes again. Therefore, the chaining method is better.

The hash function I used assigns every character a unique value and sums all the values of the characters in the key string parameter passed to the hash function.

I chose the used hash function because it makes use of all the possible characters. Therefore any possible string name will have a sum. Moreover, I found out by doing research that this is a conventional way for using hash functions with a key string.

References used for learning purposes only:

- Code on blackboard
- <https://www.geeksforgeeks.org/implementing-hash-table-open-addressing-linear-probing-cpp/>

Note:

I used only the names in my tables recalling that you mentioned in the lab that it is okay to only put the names not the class Employee itself.