PFII Lab 14: Hash Tables

In this lab I learned about the implementation of Hash table containers using key arrays and two methods of hashing (multiplication and division). Something interesting that I observed during this lab was the process of both overflow and collision (where two different keys result in the same hash value). This lab helped show an application of programming and data storage as the Hash table is a method of encryption where string keys are converted into hash values to be stored separately. This lab also provided good practice with ASCII character values and their implementation in methods dealing with strings. Comparing the content of the lab to password storage for websites and such was a good application. Another thing I learned in this lab was the importance of variable types when dealing with integer and decimal values. When separating the fractional and whole # components of d in the multiplication hashing method, I had to cast (unsigned int) rather than (int) to prevent overflow and an absurd number for the Hash index.

Division Method:

|  |  |  |
| --- | --- | --- |
| **# of keys** | **# of collisions** | **Max length of linked list** |
| 25,000 | 2634 | 5 |
| 50,000 | 8811 | 6 |
| 75,000 | 16982 | 7 |

Multiplication Method:

|  |  |  |
| --- | --- | --- |
| **# of keys** | **# of collisions** | **Max length of linked list** |
| 25,000 | 2588 | 5 |
| 50,000 | 8895 | 6 |
| 75,000 | 17055 | 7 |

*\*For both the tables above, the # of linked lists would be equal to the # of collisions for each run*





