Using the skeleton app provided in the ***/Interesting Stuff/Hash Tables – Open Addressing/***folder, complete the implementation of an *Open Addressing* collision resolution hash table.

Use the provided input files (100.data, 1000.data etc.) to feed your hash table.

Requirements:

* Implement all the operations as described in the skeleton
* At the end, after every element has been inserted, make sure to print out the maximum number of collisions which occurred.
* Also at the end compute the number of times the hash table was resized
* Play around with the hash function to see which one works better (less collisions)
* Play around with the fill factor and initial hash table size and complete the tables below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Hash Function (char \* c, int i);** | **ISF \*** | **MFF \*\*** | **Max Collisions** | **Number of Resizes** |
| H1 | .25 | .60 | 7, 34, 852, 4848, 9837, 24850, 49696, 99642 | 3 |
| H1 | .20 | .75 | 15, 34, 852, 4848, 9837, 24850, 49696, 99642 | 3 |
| H1 | .20 | .80 | 15, 34, 852, 4848, 9837, 24850, 49696, 99642 | 3 |
| H1 | .25 | .83 | 16, 67, 852, 4848, 9837, 24850, 49696, 99642 | 3 |
| H1 | .25 | .85 | 16, 102, 852, 4848, 9837, 24850, 49696, 99642 | 3 |
| H1 | .50 | .90 | 27, 114, 852, 4848, 9837, 24850, 49696, 99642 | 2 |
| H2 | .25 | .60 | 7, 13, 11, 20, 32, 25, 31, 38 | 3 |
| H2 | .20 | .75 | 33, 25, 32, 32, 53, 62, 56, 87 | 3 |
| H2 | .20 | .80 | 33, 28, 50, 38, 94, 62, 86, 100 | 3 |
| H2 | .25 | .83 | 32, 49, 52, 208, 110, 118, 122, 184 | 3 |
| H2 | .25 | .85 | 33, 49, 52, 208, 110, 144, 122, 201 | 3 |
| H2 | .50 | .90 | 47, 146, 83, 154, 233, 222, 236, 291 | 2 |
| H3 | .25 | .60 | 5, 7, 9, 12, 13, 17, 17, 17 | 3 |
| H3 | .20 | .75 | 8, 23, 19, 22, 24, 33, 34, 31 | 3 |
| H3 | .20 | .80 | 8, 23, 19, 22, 24, 33, 34, 36 | 3 |
| H3 | .25 | .83 | 20, 19, 25, 28, 34, 35, 50, 40 | 3 |
| H3 | .25 | .85 | 20, 24, 25, 29, 36, 48, 50, 59 | 3 |
| H3 | .50 | .90 | 36, 31, 34, 48, 50, 61, 57, 79 | 2 |

\* ISF = Initial Size Factor

\*\* MFF = Max Fill Factor

Have other combinations in mind? Feel free to fill-up the table with more tries to see if any interesting results come up!

H1:

***int hashFunction(char \* content, int i)***

***{***

***int length = strlen(content);***

***int k, sum;***

***for (sum=0, k=0; k < length; k++)***

***{***

***sum += content[k];***

***}***

***return (sum+i) % size;***

***}***

H2 – H3 🡪 your choices!

* Try to improve each time

Deadlines:

30411 – 18.05.2015 (before 12:00 if you want review and before 23:59 if you want a grade)  
30414 – 19.05.2015 (before 12:00 if you want review and before 23:59 if you want a grade)