POTD39.1. Problem of the Day #39

Download and Extract

An initial setup of files is provided to you via a shell script: [Download potd-q39]

Using a terminal, extract the initial files by running the shell script you just downloaded (you will need to navigate to the directory where you saved the file):

```
sh potd-q39.sh
```

Your files for this problem will be in the `potd-q39` directory.

The Problem

Task 1:

Complete the function `simpleHashFunction` in `Hash.cpp` that accepts a string and the size of the hash table `int M`. The function will take the string, performs some computation on it, and then returns an index in the range 0 to $M-1$. For this POTD, use a simple implementation: sum the ASCII values of the letters in the string.

If the hash table size M is small compared to the resulting summations, then this hash function should do a good job of distributing strings evenly among the hash table slots, because it gives equal weight to all characters in the string. This is an example of the *folding* approach to designing a hash function. Note that the order of the characters in the string has no effect on the result.

Task 2:

Now write the function `int countCollisions (int M, vector<string> inputs)`. It takes the size of the hash table as input and a vector of strings. It goes over all the input strings, performs the above `simpleHashFunction` over it and counts the total number of collisions at each index in the hash table (index ranges from 0 to $M-1$). In the main function, there is a code to test how the number of collisions change on chaging the size of the hash table. We vary the size from 1 to 11.

Upload Solution

Drop files here or click to upload.

Only the files listed below will be accepted—others will be ignored.

Files

○ Hash.cpp
not uploaded

Save & Grade | Save only

POTD 39

Total points:	0/1
Score:	0%

Question

Value:	1
History:	
Awarded points:	0/1

Report an error in this question

Previous question

Next question