* Counting Code Lines
* **Compile the project:**

Navigateto CountingCodeLines root folder and type into:

npm install

After installation succeed, Execute the program:

babel-node src/index.js test/testfiles/test1.java

src/index.js is program entry point.

Test/testfile/test1.java is the file to be tested.

* **Screenshot of Output**



**C) run tests**

npm run test



We are using mocha + chai to do unit test.

\*Bonus: Run npm run performance to see each function’s performance.

* BloomFilter
* **Compile the project:**

Navigateto CountingCodeLines root folder and type into:

npm install

After installation succeed, Execute the program:

babel-node src/index.js src/dictionary/wordlist.txt

src/index.js is task1’s entry point.

src/dictionary/wordlist.txt is the dictionary

* **Screenshot of Output**



**Task 2:**

Type in:

babel-node src/task2.js

src/task2.js is task 2’s entry point

* **output:**

Given dictionary size: 338883:

|  |  |  |  |
| --- | --- | --- | --- |
| False Positive rate | BitBuffer Size | Hash functions | Random words length |
| 0.23% | 1234567 | 5 | 1000 |
| 0.5% | 1234567 | 10 | 1000 |
| 1% | 1234567 | 50 | 1000 |
| 0.51% | 1234567 | 10 | 10000 |
| 0.0% | 12345678 | 10 | 1000 |
| 0.0% | 12345678 | 10 | 10000 |
| 1.0% | 1234 | 10 | 1000 |



Example: Bitbuffer size 12346 hash function 9 words 10000

Run test:

Npm run test



Performance test:

Npm run performance.