UNIVERSITY OF CALGARY
DEPARTMENT OF COMPUTER SCIENCE
Winter 2021
CPSC 355: Computing Machinery I

Assignment 01

Weight: 8.33% of the final grade
Due: Jan 31st (11:59 PM)

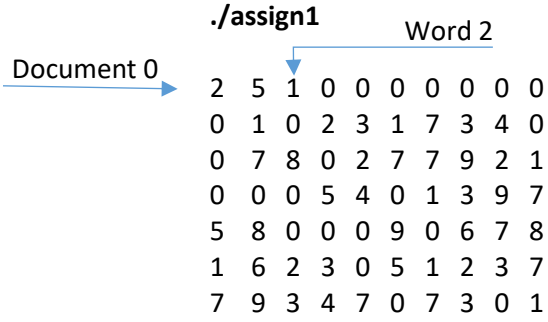
Overview

1. Your program will emulate a search engine. It reads a table of integers from a file **or** randomly creates the table and displays it to the user. The rows of this table represent text documents, and the columns represent words that may appear in these text documents. Each cell with coordinates [*i,j*] in the table contains the number of occurrences of word *j* in document *i*.
2. Your program returns a list of the top *n* documents containing word *j*. The user specifies both *n* and *j*.

Details

Display to the user a 7×10 table of random/given positive integers between 0 and 9. The user can optionally provide a file name containing an existing table, if applicable, using a command-line argument, such as (if occurrences are randomly generated):

./assign1




```
2 5 1 0 0 0 0 0 0 0
0 1 0 2 3 1 7 3 4 0
0 7 8 0 2 7 7 9 2 1
0 0 0 5 4 0 1 3 9 7
5 8 0 0 0 9 0 6 7 8
1 6 2 3 0 5 1 2 3 7
7 9 3 4 7 0 7 3 0 1
```

In this example, there are 7 documents and 10 possible words. Document 0 contains 2 occurrences of word 0, 5 of word 1, 1 of word 2, and no occurrences for words 3 to 9.

If occurrences are read from a file named *occurrences.txt*, then the name of the file is also specified:

./assign1 occurrences.txt

```
9 1 0 0 0 0 7 3 4 0
1 3 5 8 5 3 0 5 4 0
7 3 9 0 0 0 5 8 0 0
0 0 0 5 4 5 3 0 5 1
0 0 1 6 3 1 4 7 0 7
2 5 1 0 0 0 0 0 0 0
5 4 0 1 2 7 7 9 2 1
```

The user then enters the index of the word s/he is searching for and the number of the top documents containing the word, as follows:

```
Enter the index of the word you are searching for: 5
How many top documents you want to retrieve? 2
```

Your program should return the documents with the top 2 highest frequencies of the searched word in index 5. For example, your program will return the following output if the attached occurrences.txt was used:

```
Document 6
Document 3
```

Frequency of word j in document i is: *occurrences[i,j] / size of document i*

The size of document *i* (*row i*) is the sum of occurrences of all of its words (sum of all columns in *row i*).

A log file must be created showing: the initial table, user input and the search result before exiting the program.

Modularity

Your code must be divided into functions as appropriate. At a minimum, you must define the following functions (we are not showing all necessary arguments):

- `initialize(*table)`
- `randomNum(m,n);` `m` and `n` are the lower and upper bounds for the random number. You can use the C library function `rand()`.
- `display(*table)`
- `topRelevantDocs(*table, n)`
- `logToFile()`

Submission

- Name your program ***assign1.c***
- Create a script file and call it ***assign1.script***. The script file shows that the program submitted runs and, therefore, it serves as proof when evaluating your submission.
- Name your log file ***assign1.log***. The log file is for documentation purposes.
- Submit a ***README.txt*** file. The readme file should provide insights into how you implemented your program (i.e., steps you followed to make your code work).
- Submit your work to the appropriate Dropbox on D2L.

Late Submission Policy

The Late policy for assignments is as follows:

- 20% grade penalty for 1 calendar days of lateness.
- 40% grade penalty for 2 calendar days of lateness.
- A grade of zero for more than 2 days of lateness.

Academic Misconduct

This assignment is to be done by individual students: your final submission must be your own original work. Teamwork is not allowed. Any similarities between submissions will be further investigated for academic misconduct. While you are encouraged to discuss the assignment with your colleagues, this must be limited to conceptual and design decisions. Code sharing by any means is prohibited, including *looking* at someone else's paper or screen. The submission of the compiler-generated assembly code is absolutely prohibited. Any re-used code of excess of 5 lines in C and 10 lines in assembly (10 assembly language instructions) must be cited and have its source acknowledged. Failure to credit the source will also result in a misconduct investigation.

D2L Marks

Marks posted on D2L are subject to change (up or down).

Marking Criteria

Code compiles	5	_____
Code runs	5	_____
Correct results	15	_____
Log file functionality	10	_____
User interface (input validation, implementing all features)	15	_____
Random numbers	10	_____
Modularity	15	_____
Command-line arguments	5	_____
Passing parameters by reference	5	_____
Script file	5	_____
Code readability (formatting documentation)	10	_____
Total	**100**	**_____**