Homework #3

A. **(20 pts) Assess the impact of the hash table size.** Set the hash table to a fixed value (**m**, see below). Set the size of your hash table (***m***) to 1 million, 10 million, 30 million, and 60 million elements.

For each of your 4 hash table sizes, how many collisions did you observe while populating the hash?

1. For the hash size 1000000 - 99000001 collisions
2. For the hash size 50000000 - 64405364 collisions
3. For the hash size 100000000 – 51588542 collisions 4. For the hash size 200000000 – 41393877 collisions.

For each of your 4 hash table sizes, how long did it take you to populate the hash table? Do the timing results make sense (provide big O notation)? Explain.

1. The time taken for the hash size 1000000 - 99.2899 seconds
2. The time taken for the hash size 50000000 - 97.9511 seconds
3. The time taken for the hash size 100000000 - 100.613 seconds
4. The time taken for the hash size 200000000 - 100.509 seconds

For all the hash table sizes, the time taken is almost similar, so the big-O notation would be **O(N)** which is linear time complexity.

ondemand.hpc.nau.edu/pun/sys/dashboard/files/fs//scratch/sd2672/homework3/partA/output1.txt

```
The number of arguments passed: 5
The first argument is: /scratch/sd2672/homework3/./source
The second argument is: /common/contrib/classroom/inf503/human_reads_trimmed.fa
The third argument is: /common/contrib/classroom/inf503/genomes/human.txt
The fourth argument is: partA
The fifth argument is: 1000000
hash table size - 1000000
The total no of queries is 100000000
The time taken to populate the hash table of size 1000000 is 99.2899 seconds

The total no of collisions while populating the hash are 99000001
```

ondemand.hpc.nau.edu/pun/sys/dashboard/files/fs//scratch/sd2672/homework3/partA/output2.txt

```
The number of arguments passed: 5
The first argument is: /scratch/sd2672/homework3/./source
The second argument is: /common/contrib/classroom/inf503/human_reads_trimmed.fa
The third argument is: /common/contrib/classroom/inf503/genomes/human.txt
The fourth argument is: partA
The fifth argument is: 50000000
hash table size - 50000000
The total no of queries is 100000000
The time taken to populate the hash table of size 50000000 is 97.9511 seconds

The total no of collisions while populating the hash are 64405364
```

ondemand.hpc.nau.edu/pun/sys/dashboard/files/fs//scratch/sd2672/homework3/partA/output3.txt

```
The number of arguments passed: 5
The first argument is: /scratch/sd2672/homework3/./source
The second argument is: /common/contrib/classroom/inf503/human_reads_trimmed.fa
The third argument is: /common/contrib/classroom/inf503/genomes/human.txt
The fourth argument is: partA
The fifth argument is: 100000000
hash table size - 100000000
The total no of queries is 100000000
The time taken to populate the hash table of size 100000000 is 100.613 seconds

The total no of collisions while populating the hash are 51588542
```

ondemand.hpc.nau.edu/pun/sys/dashboard/files/fs//scratch/sd2672/homework3/partA/output4.txt

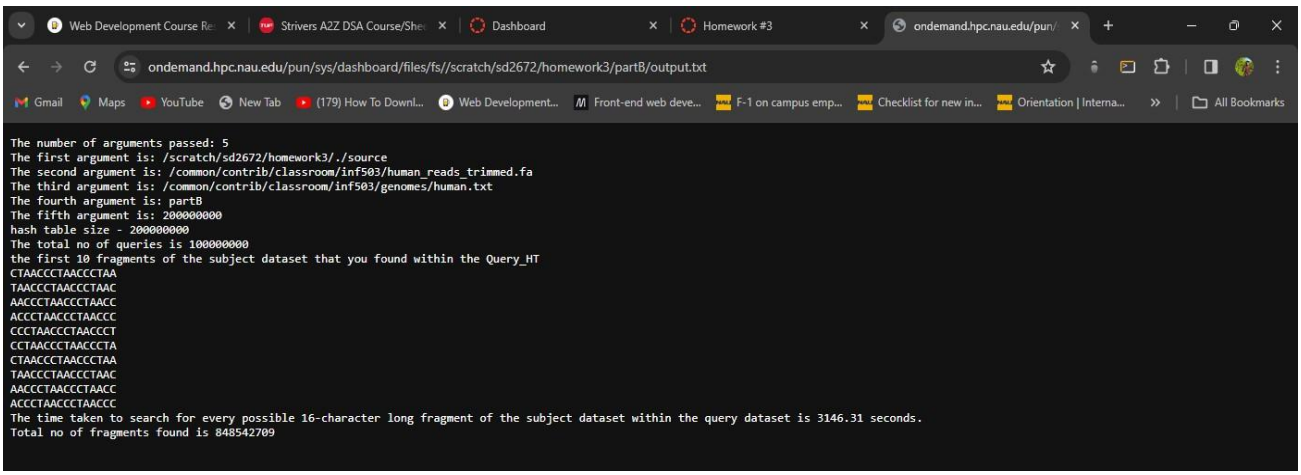
```
The number of arguments passed: 5
The first argument is: /scratch/sd2672/homework3/./source
The second argument is: /common/contrib/classroom/inf503/human_reads_trimmed.fa
The third argument is: /common/contrib/classroom/inf503/genomes/human.txt
The fourth argument is: partA
The fifth argument is: 200000000
hash table size - 200000000
The total no of queries is 100000000
The time taken to populate the hash table of size 200000000 is 100.509 seconds

The total no of collisions while populating the hash are 41393877
```

B. (20 pts) Searching speed: Set the hash table size to 60 million. Populate the hash table with the sequence fragments from the *query dataset*. Read the entire *subject dataset* into a single, concatenated character array (same way you did it in HW#1). Implement a search function which would search for 16character fragments of the subject sequence within the Queries_HT object. Iterate through all 16character long fragments of the *subject dataset*, searching for each one in the *query dataset*.

- How long did it take to search for every possible 16-character long fragment of the *subject dataset* within the *query dataset*?

  3146.31 seconds
- How many such fragments did you find?

  848542709
- Print the first 10 fragments of the *subject dataset* that you found within the Query_HT.



Steps of execution:

- Created total of three files main.cpp, header_definitions.cpp and header.h
- Header.h file contains all the header files that are used in the program
- The main.cpp contains the main function and all the function calls required to get desired output
- Header_definition.cpp file contains all the function definitions which are declared in the header file
- Created a make file to run the code
- Uploaded all the above files to a directory on monsoon
- There I have opened terminal and entered the command "make" then source executable file is generated next we need to run the source file with file path and the part of execution.
- The command for execution of part A is
  ./source /common/contrib/classroom/inf503/human_reads_trimmed.fa
  /common/contrib/classroom/inf503/genomes/human.txt partA (hashsize)
- The command for execution of part B is
  ./source /common/contrib/classroom/inf503/human_reads_trimmed.fa
  /common/contrib/classroom/inf503/genomes/human.txt partB (hashsize)