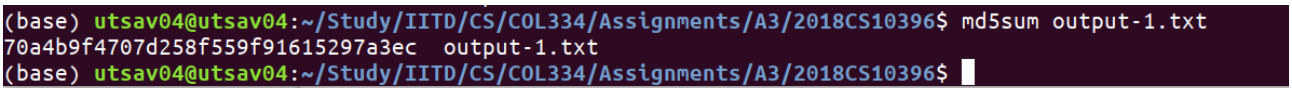
COL334 – COMPUTER NETWORKS
ASSIGNMENT 3
LARGE FILE DOWNLOAD

UTSAV DEEP
2018CS10396

PART 1 :

How to run?
$ javac q1.java
$ java q1 > output-1.txt
$ md5sum output-1.txt

This will save the output in the file named output1.txt and the md5sum will be checked and result will be either matches or doesnot match depending upon whether the md5sum matched or not.



```
(base) utsav04@utsav04:~/Study/IITD/CS/COL334/Assignments/A3/2018CS10396$ md5sum output-1.txt
70a4b9f4707d258f559f91615297a3ec  output-1.txt
(base) utsav04@utsav04:~/Study/IITD/CS/COL334/Assignments/A3/2018CS10396$ 
```

Explaining the code:
-> In this part, I first opened socket connection to “vayu.iitd.ac.in” on port number 80 which is the port number for http.
-> Then using the getOutputStream() method, I created a way send messages to the server.
-> Similarly using the getInputStream() method, I a stream for all the messages from the server to reach me (the client)
-> Then I wrote the GET request in a string as follows:

```
sentence = "GET /big.txt HTTP/1.1\r\nHost: vayu.iitd.ac.in\r\nConnection: keep-alive\r\n\r\n";
```

-> This sentence was written onto the output stream byte by byte.
-> Then, response of the server was read byte by byte using the .read() method of BufferedReader.
-> The characters were read untill the whole stream was read and the characters were output to the file name “output1.txt” which is the ouptut file.
-> Then the md5sum was checked and depending upon the result, necessary output was shown.
-> Then the socket was closed.

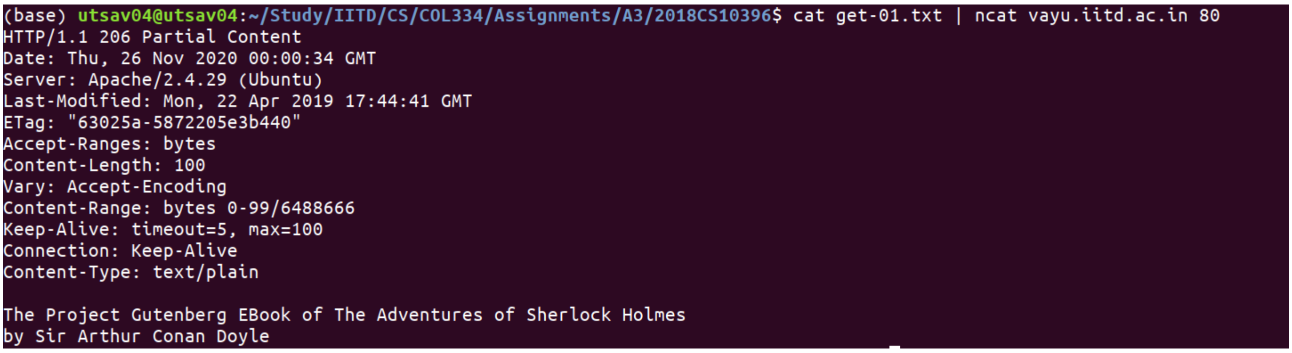
-> This is how part 1 was done.

PART 2 :

How to run?
$ javac q2.java
$ java q2 > output-2.txt

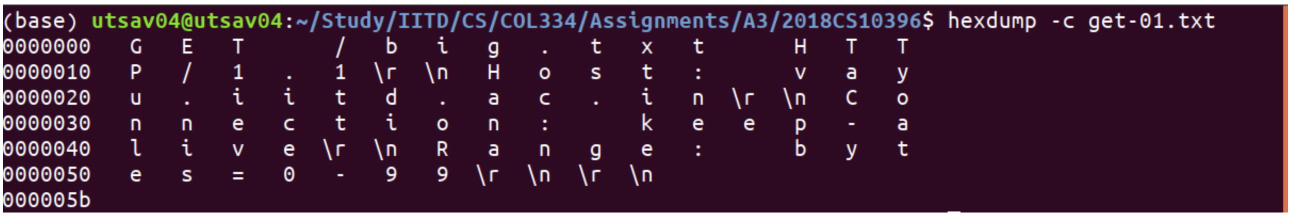
In this part, I added the range field to the GET request and the GET request became:
"GET /big.txt HTTP/1.1\r\nHost: vayu.iitd.ac.in\r\nConnection: keep-alive\r\nRange: bytes=0-99\r\n\r\n"

I also made changes to the get-01.txt file using vim. The file get-01.txt has been added to the solution directory.



```
(base) utsav04@utsav04:~/Study/IITD/CS/COL334/Assignments/A3/2018CS10396$ cat get-01.txt | ncat vayu.iitd.ac.in 80
HTTP/1.1 206 Partial Content
Date: Thu, 26 Nov 2020 00:00:34 GMT
Server: Apache/2.4.29 (Ubuntu)
Last-Modified: Mon, 22 Apr 2019 17:44:41 GMT
ETag: "63025a-5872205e3b440"
Accept-Ranges: bytes
Content-Length: 100
Vary: Accept-Encoding
Content-Range: bytes 0-99/6488666
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/plain

The Project Gutenberg EBook of The Adventures of Sherlock Holmes
by Sir Arthur Conan Doyle
```



```
(base) utsav04@utsav04:~/Study/IITD/CS/COL334/Assignments/A3/2018CS10396$ hexdump -c get-01.txt
0000000   G   E   T       /   b   i   g   .   t   x   t       H   T   T
0000010   P   /   1   .   1  \r  \n   H   o   s   t   :       v   a   y
0000020   u   .   i   i   t   d   .   a   c   .   i   n  \r  \n   C   o
0000030   n   n   e   c   t   i   o   n   :       k   e   e   p   -   a
0000040   l   i   v   e  \r  \n   R   a   n   g   e   :       b   y   t
0000050   e   s   =   0   -   9   9  \r  \n  \r  \n
000005b
```

PART 3 :

How to run?
$ javac q3.java
$ java q3 > output-3.txt
$ md5sum output-3.txt

-> Multiple threads were made and and each made a connection to the host.
-> I have used dynamic allocation of chunks using a synchronized object which will tell each thread , which chunk is avaiable for download.

-> Necessary bookkeeping has been done by the synchornised object.
-> An arraylist maintains all the parts which have been found, at combines them in the end.
Q) Does your download time keep decreasing with more and more parallel TCP connections?
-> Yes the download time decreases with more and more parallel TCP connections.
-> A significant decrease is seen with increasing connections.

Number of threads	Time taken
1	2 min
4	50 sec
20	25 sec
60	14 sec

-> Yes, some connections are faster than the others. A few theads are downloading far more bytes than the rest.

Q) Spread your connections between vayu.iitd.ac.in and norvig.com :
-> For spreading the connections between vayu.iitd.ac.in and norvig.com, in alloted 30 threads to each.
-> However, this increased the time taken to download to 25 sec.
-> This is because, norvig.com server is slower.
What does this tell you about where bottlenecks lie? Is one server faster than the other?
-> The bottleneck in this case is the norvig server.
-> Vayu.iitd.ac.in server is faster than the norvig.com server.
Is your program able to use this to download more from the faster server?
-> Yes, my program is able to use this to download more fromthe faster server becuase i have dynamically allocated chunks to the threads.

-> In general , dividing into multiple servers increases overall server usage and helps reduce too much load on a single server.
-> In these cases, the server which is the slowest will serve the requests slowest and thus will act as bottleneck link.

PART 4 :

How to run?

```
$ javac q4.java
$ java q4 > output-4.txt
$ md5sum output-4.txt
```

-> In this part I have made changes to the catch portion.
-> The program catches
any exception raised and keeps on trying to reconnect till it succeeds.
-> After it reconnects it proceeds as before, else it waits.
-> I have set the timeout to 10 milliseconds.

THANKYOU!