Things left to do:

1-Simultaneous connections(total no. of pipelines i.e. threads = simultaneous connections)

2-cli input format( resume or not, num\_of\_files to download, num\_of\_pipelines= num\_of\_threads) , metric\_interval\_in\_seconds, web\_address1, web\_address2, ... web\_addressn, file\_location\_to\_save1, file\_location\_to\_save2, ... file\_location\_to\_saven)

3- using byte ranges specify each thread's range and write file and combine the file a the end. also add error detection if multiple threads are not supported

4- Also do multiple downloads by splitting web\_addreses from the input and applying them through a for loop to the download function.

5- resume file by checking in the output directory how many bytes have been downloaded and start after this range of bytes if the download flag is specified in the input cli.

6- metric reporting: Necessary metrics are: download speed , (download\_bytes/totalbytes) i.e. --> percentage downloaded by a single thread. Specify these metric every i seconds obtained through cli.

Also at the end report of the total connection <downloaded\_bytes>/<total\_bytes>, download speed: <speed>kb/s