#Useage of the R scripts
1. Run project1a.r
2. Run project1b.r

#Algorithm
#For project1a
1. The project1a.r contains the webcrawler build using library function of r.
2. The crawler first goes to the main directory page and fetches the links to the papers submitted each year and stores it in a vector.
3. After all the links to the papers from page 1 is obtained, the crawler visits all the links in the vector and generates the html file of each page.
4. Once all the html pages using the links in page 1 is done, the crawler moves to the next page.
5. Again it fetches all the links to the papers submitted each year and stores the links in a vector.
6. After all the links have been added to the vector the crawler visits them one by one and generates the html files of the papers.
7. After writing all the html files for the html pages that contain the link to the papers and the paper itself the crawler terminates.

#For project1b
1. Now the system contains all the html files of the papers.
2. Functions are created using regular expressions, string functions etc to fetch the required data
3. All the data from a page is stored in a dataframe.
4. The final dataframe contains the data from all the papers.
5. The final dataframe is used to write the output file.

#Contributions
1. Ankit was responsible to create the regular expressions needed to obtain the data and writing the dataframe into the output file.
2. Sneha was responsible to create the functions that will return the requested data and generating the required dataframes
3. Abhishek was responsible for creating the webcrawler.

#Major Challenges
1. Multiple instances of the tags containing the required data.
2. Difficulty in creating the required regular expressions.
3. Formatting the substring obtained after using the regular expressions.
4. Crawler visiting useless links.
5. Finding the required links from a page so that the crawler visits them only.
6. Fetching the body of the paper as the website semantics are unorthodox.