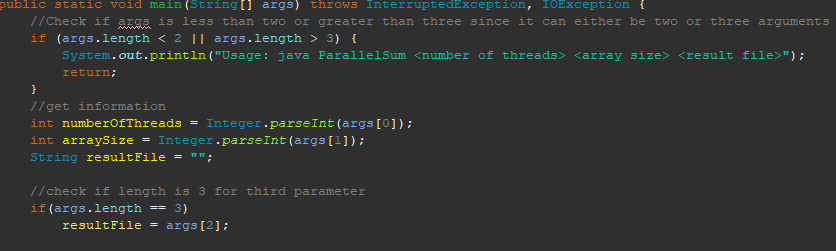
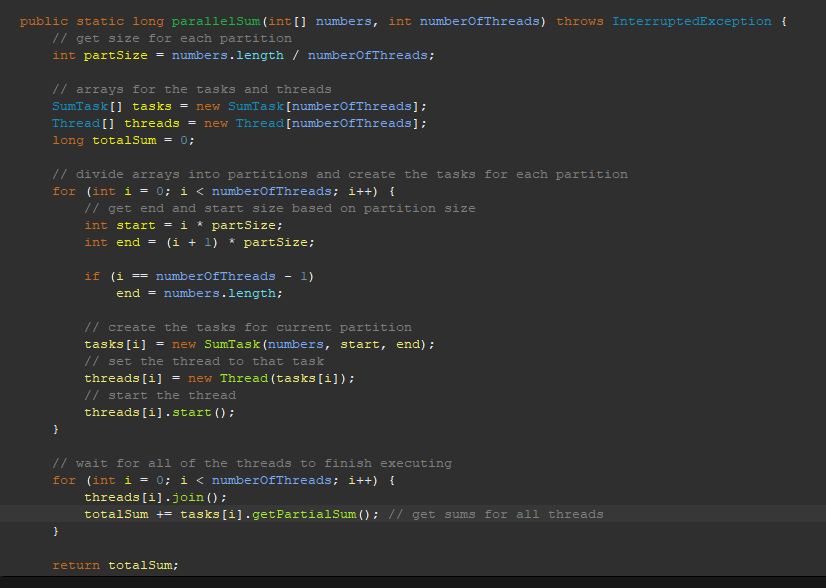
HOMEWORK WRITE UP  
This program compares two types of sum methods, one using partitions (slitting up the work using threads) and the other doing it sequentially. Here is how it works.  
  
MAIN METHOD:  
The main method starts by seeing if the size of args is less than two or more than 3 because we have an optional parameter. So, if the args is two or three the program will run regularly. Once it does that it takes in the parameters and considers the optionality of the three parameters.



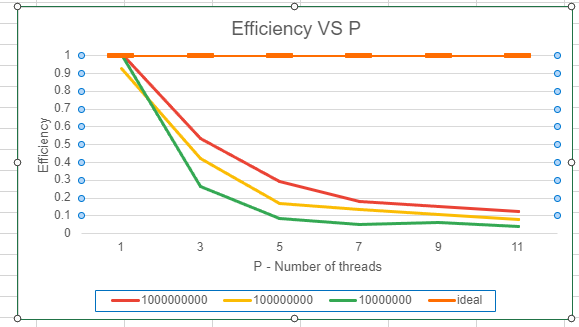
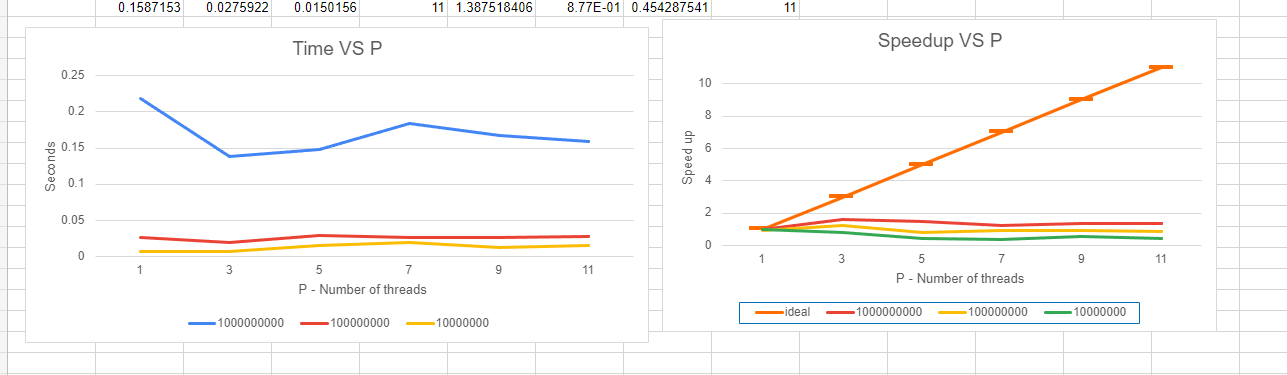
Next, we call the sequential method and calculate how long it takes to execute. Next, we call the parallelSum method and calculate the time that takes. Then we calculate the efficiency and how much it sped up. After the program checks the args length again to see if the third criteria are met. If it is met, we create a file object with the file name, check to see if it exists in the directory or not and then write the necessary information to the file. If the file exists already then, the program will not write the header to the file and keep writing the data because we set append to true in the file writer.



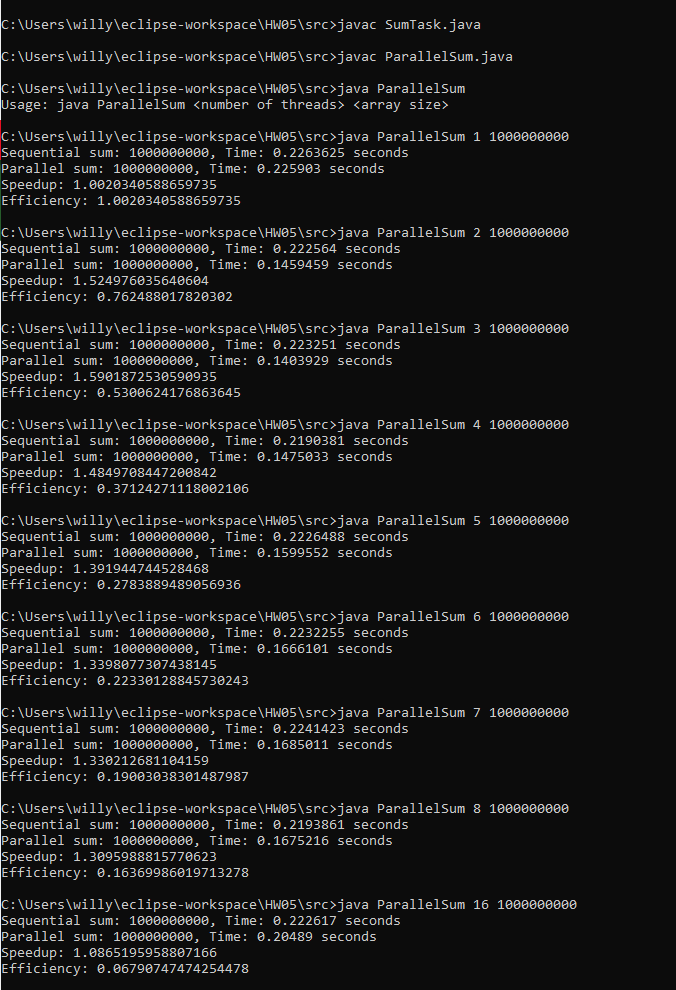
PARALLELSUM METHOD:  
This method takes in the numbers array and the number of threads. First, we must calculate how we will split the tasks. The variable “partSize” is initialized to the number's length divided by the number of threads. After that we create an array for the tasks which is just a summation object that sums the portion of the input array. We also create an array for all the threads. After that we loop through the number of threads and calculate the start and end size of each partition. After that we populate the tasks list of sum tasks and then populate the thread array with that task. Then we set the thread to start. After that loop, there is another loop to join all the threads, so they finish and get the total sum of all the tasks.   


SEQUENTIALSUM Method:  
This method takes in an array and adds all of the values in that array and then returns the sum.

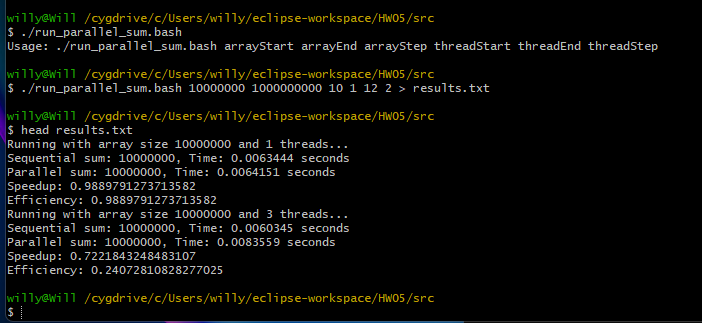
Here are the graphs

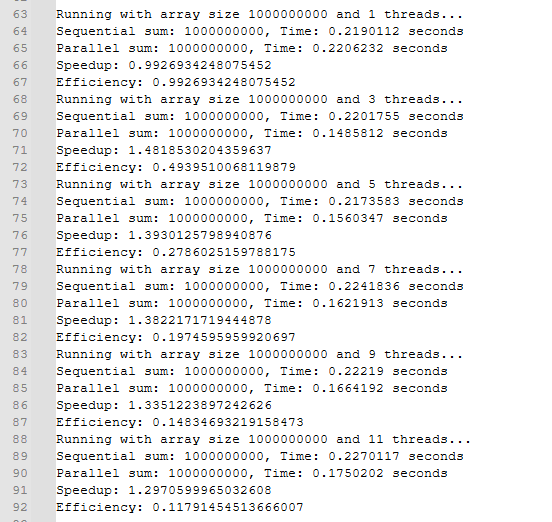
HOMEWORK SCREEN SHOTS

BEFORE EDITING PROGRAM:  


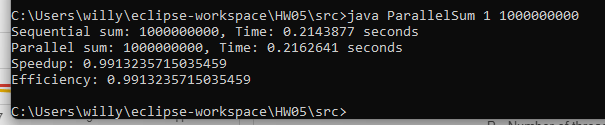
Making sure bash file works

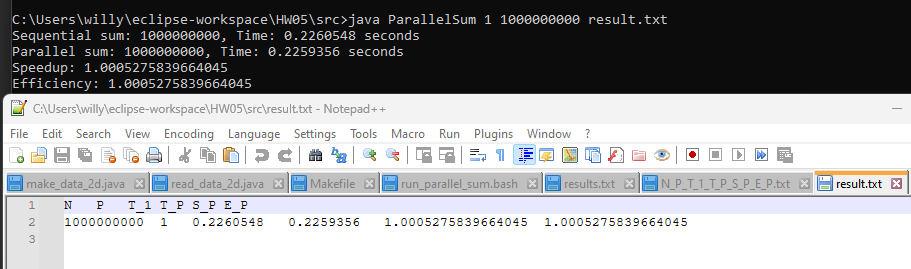
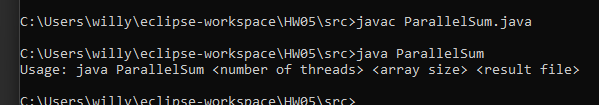


Showing results of bash



After writing program:





Showing the same display of outputs using the script