Name: Vishal Vijay Devadiga

Roll Number: CS21BTECH11061

Code Flow

- Opens input file in main and gets values of n,k
- Create k threads that execute the threadsolve function. Parent stores the threadID of the threads in an array.
  - Child Thread opens a log file to print the output of its calculations and an array on heap to store the numbers
  - The thread finds perfect numbers among the numbers assigned by the formula: $i \times \frac{n}{k} + threadNo$ where i is the i^{th} iteration of the loop in the thread and threadNo is the number of the thread(Not threadID).
  - The thread then prints the perfect numbers found to the heap array and return the pointer to the array.
- While the threads are executing the threadsolve function at their pace, parent thread is in a loop that waits for a specific thread(identified by the threadID stored)
- If that thread finishes executing(exits), then parent thread accesses the array created by the child thread and then reads all the perfect numbers produced by the child.
- The parent thread prints this to the output file, and repeats the procedure for all of the children threads.

Reason for allocation

The allocation of numbers to calculate is based on the formula $i \times \frac{n}{k} + threadNo$ where i^{th} iteration of the loop in the thread and threadNo is the number of the thread(Not threadID). This balances the load between the threads, compared to the sequential allocation(1 to $\frac{n}{k}$ for thread 1, $\frac{n}{k} + 1$ to $\frac{2 \times n}{k}$ for thread 2 and so on).

Some Observations

Printing the log files takes the most amount of time in the program. The actual program without printing the log files takes a lot less time.
Increasing the number of threads after a certain amount actually increased the execution time of the program.