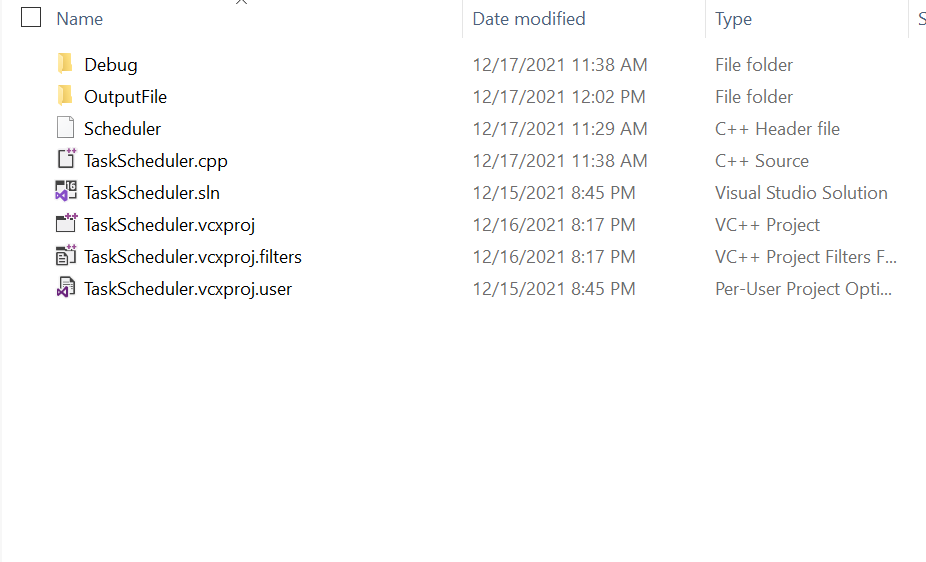
**Operating Systems Project Report**

Description

This program simulates CPU task scheduling algorithms. For each algorithm there is a set of data points that are outputted onto a “.txt” file. Each file contains randomly generated CPU times and priority levels.

Layout

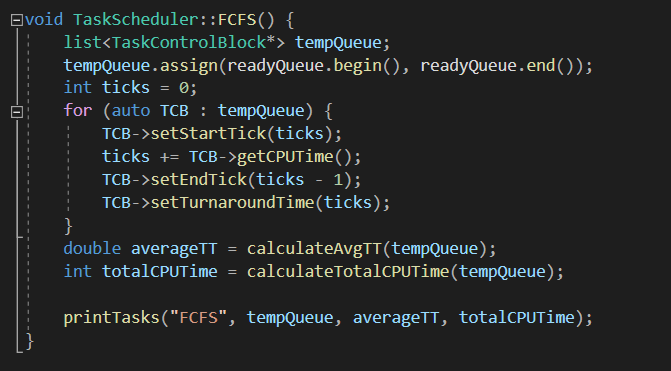
The task\_scheduler.zip folder contains three items: header file (Scheduler.h), cpp file (TaskScheduler.cpp), and an “OutputFile” folder. Each time the program runs, the data will be outputted to a file that the user specifies in the “OutputFile” folder.



Features

**FCFS**

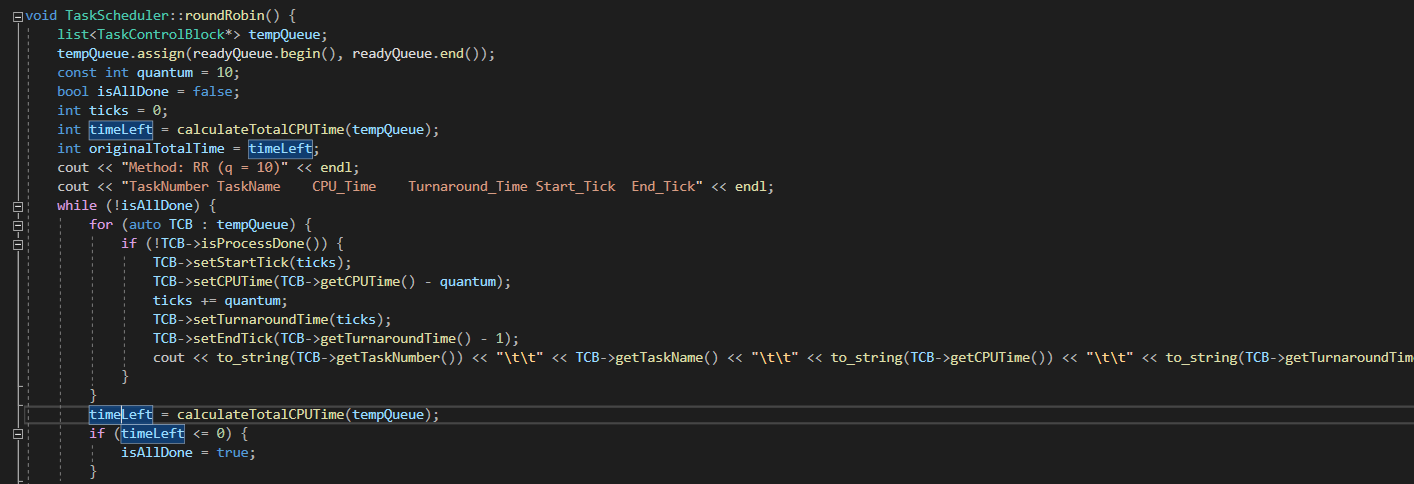
FCFS runs the algorithms in order based on arrival time. However, considering that all 10 tasks were added to the ready queue at the same time, there is no arrival time. Instead the program runs the tasks in order based on task number.



The program then outputs the data to the screen and into the specified file that the user requested at the beginning of the program.

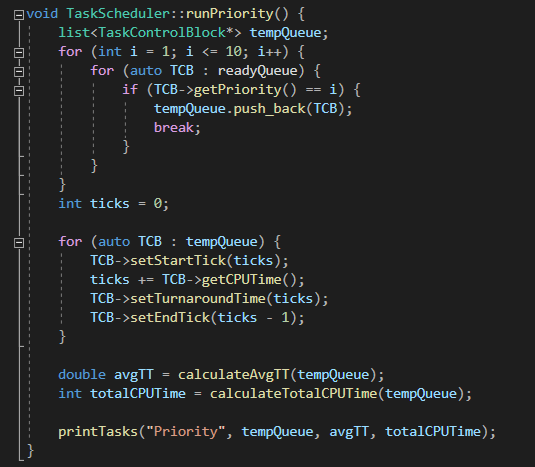
**Round Robin**

The round robin algorithm assigns a constant time quantum. Each time the program runs a task, the time quantum is subtracted from the CPU burst time. When a task’s CPU time is equal to 0, then it is considered finished and no longer runs.



**Priority**

At the beginning of the program, in the constructor, a random priority is set amongst the tasks. These are random numbers from 1 to 10. The priority algorithm adds tasks to the tempQueue in the order of their priority level, with lower numbers having higher priority. After that the usual task system runs with each running through their entire burst time.



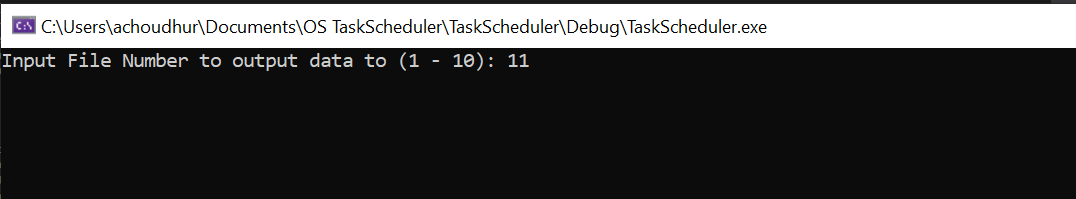
**SJF**

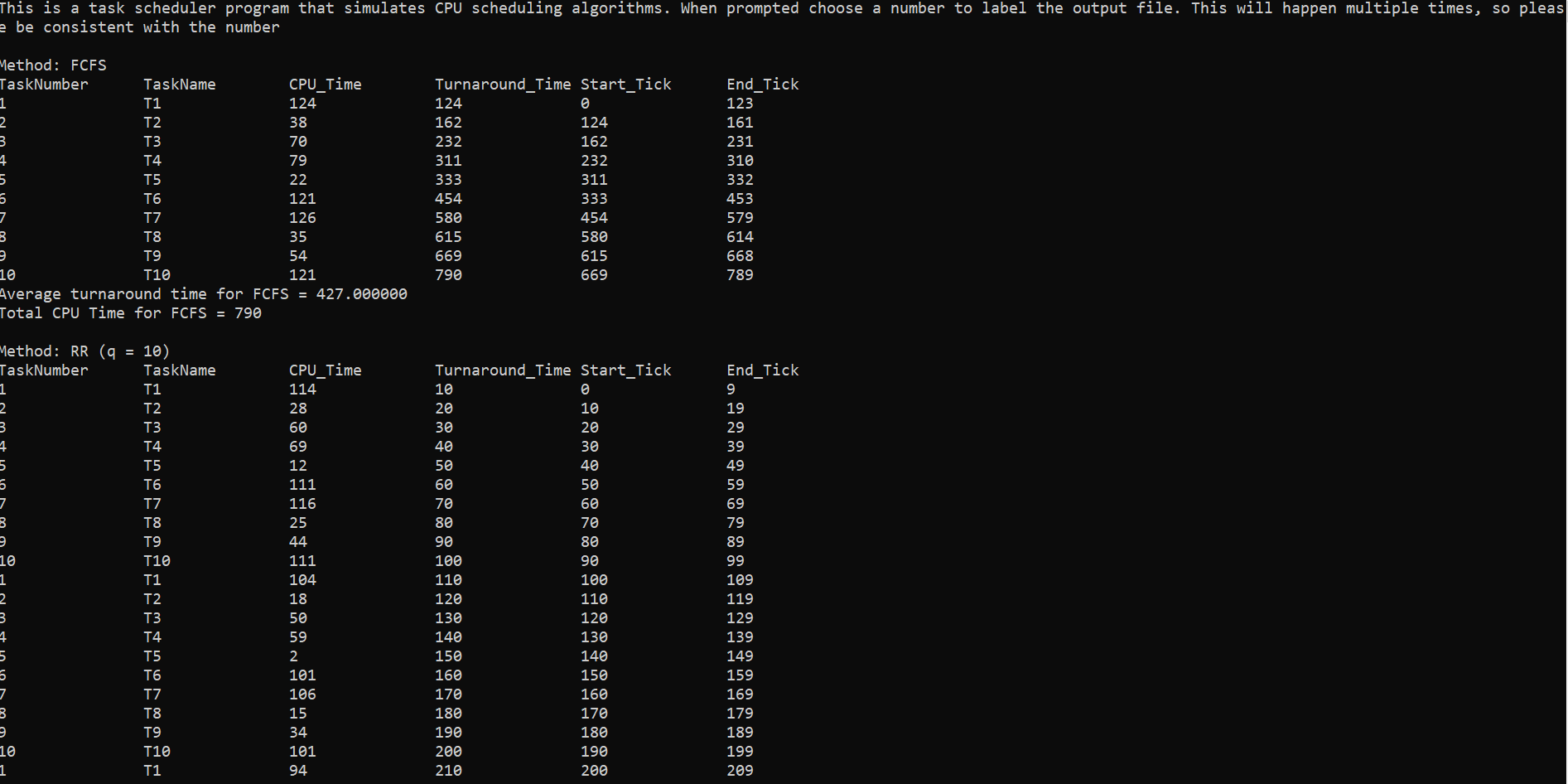
Shortest Job First runs tasks in the order of their CPU times from lowest to highest. A CPU times list is sorted out from lowest to highest and orders the tasks based on the values.

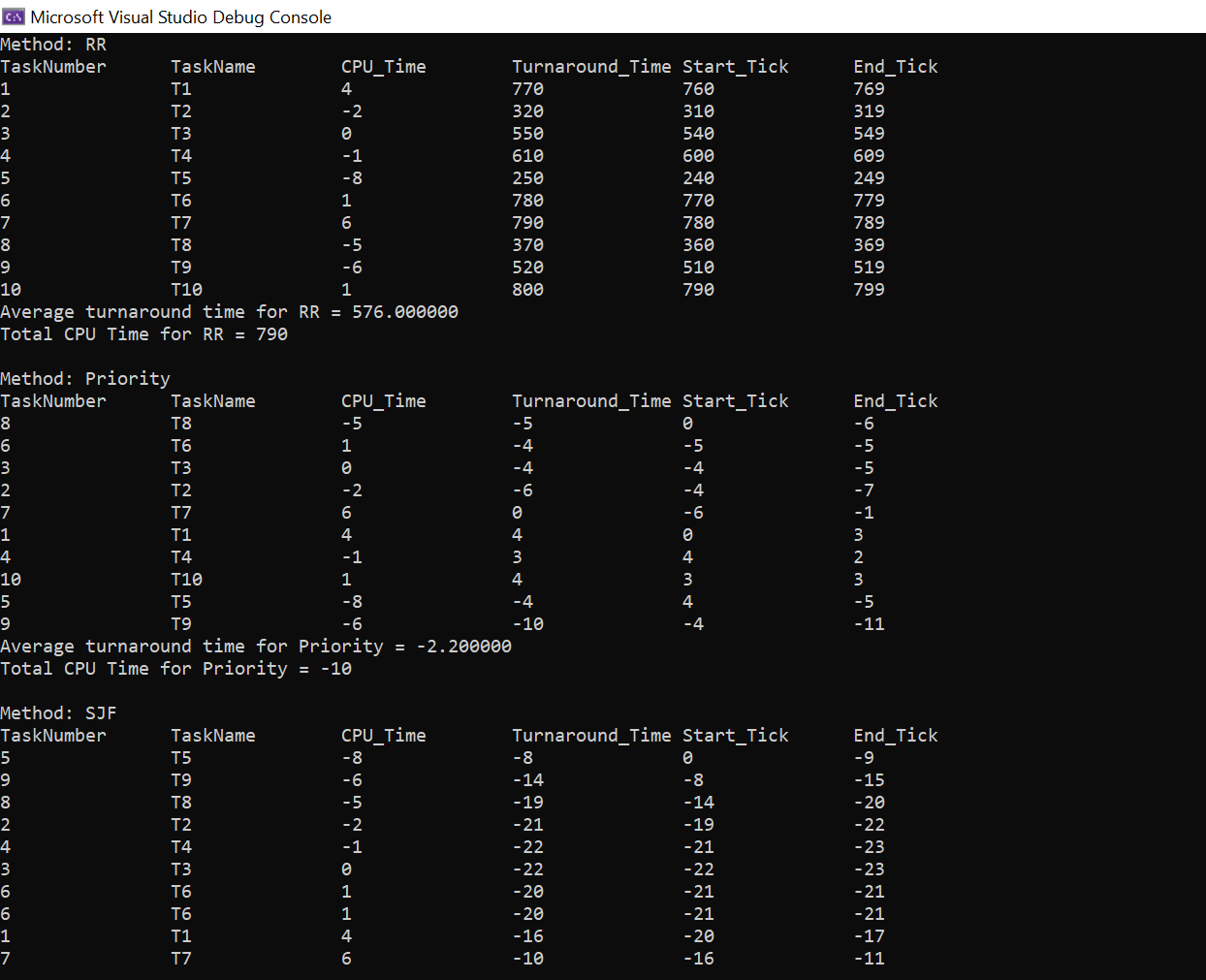


Output

The program asks for a file number that will specify which output file to output to. For example, when “1” is inputted the data outputs to the file “output\_1.txt”. From there, the algorithms run in order and output to the file after.







While the outputs are not perfect. Problems did arise when trying to only create temporary changes to the memory. With numbers turning out negative.

