

Bahri Alabey 2019400228

NOTE: Zero thread output to output1.txt, 5 thread to output2.txt 10 thread to output3.txt and optional one into output4.txt  
after make command run by "zero"  
after make command run by "five"  
after make command run by "ten"  
after make command run by "optional"

## 1 First Part

I've created a struct to give its pointer to the function. I've used this struct to give numbers array and get the results from the functions such as, minimum, maximum, range, mode, median etc. This struct approach helped me through the printing the results after finding them.

## 2 Second and Optional Part

I've put a bool array length of 10 to show whether the function that has given this struct will run that function or not. To be exact, if first element of the functions array in the struct is true, then the thread that has this struct as a parameter will run calculate minimum. I've used their orders just like in the project description.

In addition to the methods to calculate some number, I've created another function named "calculate", it'll get the threadinfo struct as an void pointer argument, convert it inside then check which functions that it'll call in a if part.

In the main part I've created an array of thread ids to give to the threads created, and create a counter for which id will be used. Also created a struct array for threads to give each thread another instance of threadInfo.

Firstly I've found the numbertodivide by dividing 10 by number of threads and round it into ceil. Rounding to ceil will make sure that all functions distributed into a thread.

Created a for loop to label which functions will work in the thread that will be created. After labeling some number of bools in the struct as true, I've created the thread and joined it. After creating a thread, I've labeled all the bools as false and proceed from the index that I've left.

At the end, I've created a file named output2.txt and write the results and time into that file.

I've realized creating some threads will reduce the time that requires to perform some operations. But after some point the overhead of creating a thread will dominate and prevent the positive effects of another thread. Especially for some small amounts of N having 10 threads will take more time than not creating another thread.