

# REPORT: Programming Assignment-2

AI20BTECH11006

## Explanation of the design

- I've declared the variables `N`, `K` as global. This was done so that I could access them from any function without the need to pass it as an argument.
- I wrote a function to check if a number is perfect or not, I did this because dividing the code into smaller sections is often better.
- to balance the load, I gave numbers at interval of `K` to each of the threads.

Example `N=10`, `K=3`

Thread1: `1`, `4`, `7`, `10`

Thread2: `2`, `5`, `8`

Thread3: `3`, `6`, `9`

- to each thread I'm passing it's thread number. Since `N`, `K` are global variables, and my balancing strategy is defined, I can choose what numbers to check for in the function `myThread` itself.
- To create file, I use `fopen` method. To get the appropriate name, I used `sprintf` to store the filename in a string which can then be passed onto the `fopen` method.
- I'm returning an array(pointer to first element) from the thread function, since I don't know the number of perfect numbers for each thread, I use a dynamic array initially of size 1. The first index represents the length of the actual array (0 initially). As and when, I get more perfect numbers, I use `realloc` to assign more memory to store numbers.
- To access the returned array(pointer), I used `pthread_join`. First, we need to convert the pointer to `void*`, because that is required by `pthread_join`
- On top, I added few error situations. Like, if  $N < 1$  or  $K < 1$ . Since, it wasn't explicitly asked for, I didn't go out of my way to do more naive error handling.