SOS_Assignment1_Task1

ID_Number: IT19022666

Name: P.T.D. Thissera

One of the main obstacles I encountered when trying to complete the code was the problem of how to read the data line by line in the dataset.txt file.

```
int read_n()//This function is created by reading the numbers in the dataset.file
{
        struct t1;
        t1.a1[0] = 0;
        int tot, t = 0;
        char chr[BUFSIZ];
        int c2;

        FILE *cfPtr;//File pointer
        cfPtr = fopen("dataset.txt","r");//Open a text file called 'dataset.txt' to read data

        if (cfPtr == NULL)//Check if the 'dataset.txt' is open properly
        {
                printf("Wrong !!\n");
                return -1;
        }

        int i;
        while (fgets(chr,sizeof chr,cfPtr) != NULL)//Reading the 'dataset.txt' file
        {
                int x;
                int y = 0;
                char *chr2 = chr;
                int c2 = 0;
                t++;
                while (sscanf(chr2,"%d%n",&x,&y) == 1)
                {
                        chr2 += y;//The FLAG
                        c2++;
                        tot = tot + x;//calculatin the summation of the each of line numbers
                }
                ++i;
```

First, I created a **read_n ()** function and then I created a file pointer and open the text file called **'dataset.txt'**. Then I put **'if '**condition to check if the file is opening properly. If the file not open, then program display a message called **'Wrong'**. After that I put while loop to read the data of the file and inside that while loop, I put another while loop to find the summation of the lines' numbers. The **chr** array's first character is pointed by the **chr2**. The function **sscanf ()** read the line using the file pointer called **'cfPtr'**, passes the integer value to field and the **'%n'** is format specifier send the numbers of blank spaces to **'y'**. Then that value (y) is added to the **chr2**, so that it skips the blank space and points to the next value. After that I put a **'FLAG'** in there. Using that '**FLAG**' we can read numbers from the dataset.txt file.

```
root@kali:~# vi task1.c
root@kali:~# gcc -pthread task1.c -o a
task1.c: In function 'main':
task1.c:43:63: warning: cast to pointer from integer of different size [-Wint-to-pointer-cast]
   43 |     checking = pthread_create(&id[t1.v],NULL, (void *)cal_avg, (void *)t1.a2[t1.v]);//Create thread using different attributes
      |                                                               ^
root@kali:~# ./a
Creating tread_1 status: OK
1 Line sum is: 150
1 Line Average is: 30.00
Creating tread_2 status: OK
2 Line sum is: 90
2 Line Average is: 10.00
Creating tread_3 status: OK
3 Line sum is: 15
3 Line Average is: 5.00
root@kali:~#
```