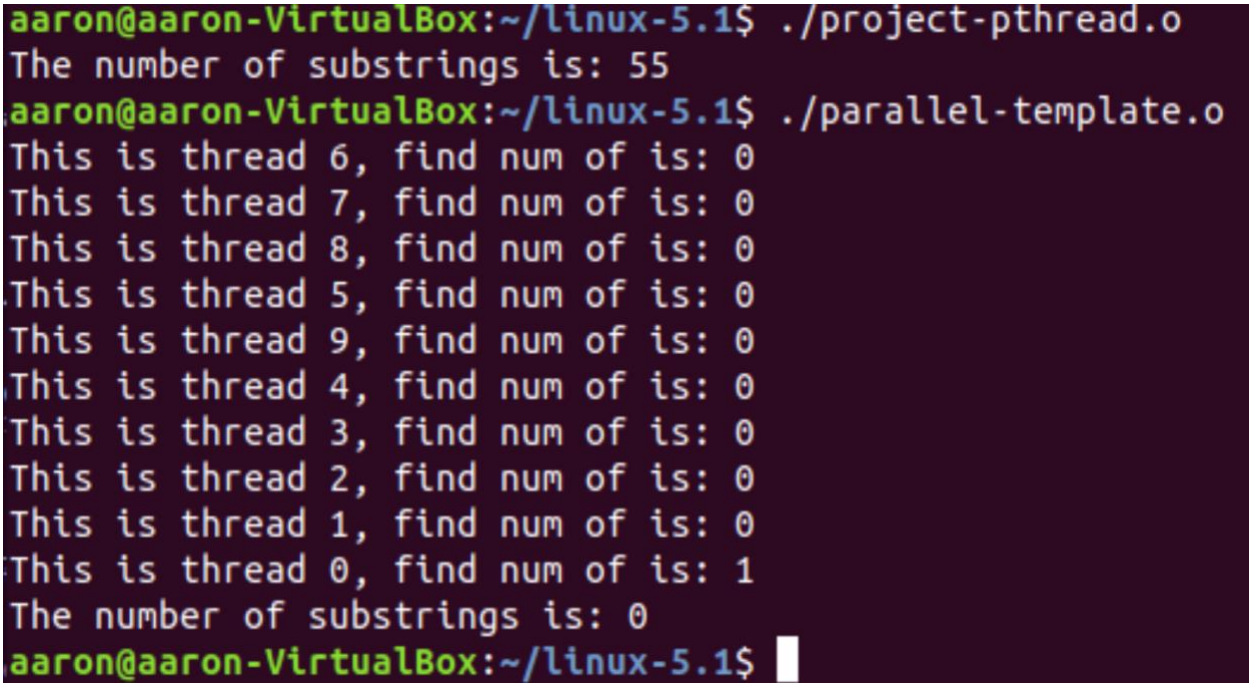
Aaron Badgett
Assignment 3

```c
#include <pthread.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

#define MAX 10240
#define NUM_THREADS  10

int total = 0;
int n1,n2;
char *s1,*s2;
FILE *fp;
int countArray[NUM_THREADS]={0};


//read input file and generate string s1/s2 and length n1/n2
int readf(FILE *fp)
{
        if((fp=fopen("strings.txt", "r"))==NULL){
                printf("ERROR: can't open string.txt!\n");
                return 0;
        }
        s1=(char *)malloc(sizeof(char)*MAX);
        if(s1==NULL){
                printf("ERROR: Out of memory!\n");
                return -1;
        }
        s2=(char *)malloc(sizeof(char)*MAX);
        if(s1==NULL){
                printf("ERROR: Out of memory\n");
                return -1;
        }
        /*read s1 s2 from the file*/
        s1=fgets(s1, MAX, fp);
        s2=fgets(s2, MAX, fp);
        n1=strlen(s1);  /*length of s1*/
        n2=strlen(s2)-1; /*length of s2*/

        if(s1==NULL || s2==NULL || n1<n2)  /*when error exit*/
                return -1;
        return 0;
}
```

```
int num_substring(int t)
{
//add your logic here
//1, how to distribute different parts of string s1 into different threads
//2, how to sun up the total number of substring fron all threads
    int i,j,k;
    int count;

    long t_id = (long)t;

    for(i=((n1 / NUM_THREADS) *t_id); i < (n1 / NUM_THREADS); i++)
    {
            count = 0;

            for (j = i, k = 0; k < n2; j++, k++)
            {

                    if (*(s1 + j) != *(s2 + k)) {break;}
                    else {count++;}
                    //printf("count: %d\nN2: %d", count, n2);
                    if (count == n2){ countArray[t_id] = countArray[t_id] + 1; }

            }
    }

    for (int i = 0; i < NUM_THREADS; i++)
            total += countArray[i];

    return total;
}

void *calSubStringThread(void *threadid){
    long tid = (long)threadid;
    printf("This is thread %ld, ", tid);
    int num = num_substring(tid);
    printf("find num of is: %d\n", num);
    pthread_exit(NULL);
}

int main(int argc, char *argv[])
{
    pthread_t threads[NUM_THREADS];
    int t, rc;
    int totalNum = 0;

        readf(fp);

        for(t=0; t<NUM_THREADS; t++){
                rc = pthread_create(&threads[t], NULL, calSubStringThread, (void *) (size_t)t);
        if (rc){
            printf("ERROR; return code from pthread_create() is %d\n", rc);
            exit(-1);
        }
    }

    for(t=0; t<NUM_THREADS; t++){
        pthread_join(&threads[t], NULL);
    }

        printf("The number of substrings is: %d\n", totalNum);
        return 1;
}
```



Output Description:

The reasoning behind my logic is very similar to the one provided. I was trying to add the thread id to a running an array of id's that I can track where and what parts of the substring matched the other string. I had an issue with 'n1' and was not reading properly.