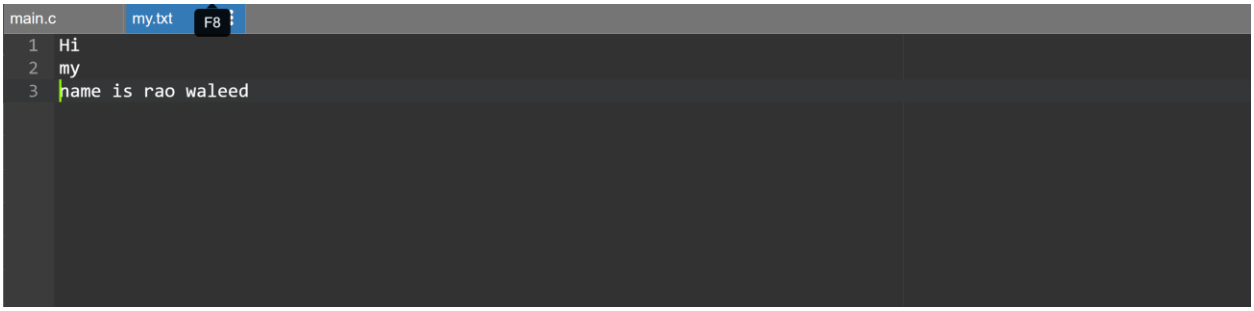
Graded lab 04

Code:

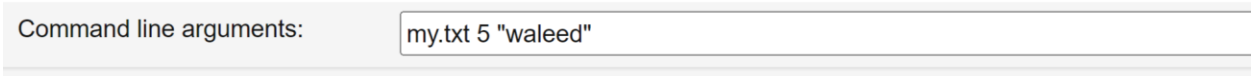
```
#include <stdio.h>
#include <stdlib.h>
#include<pthread.h>
#include <stdbool.h>
#include <unistd.h>
#include <string.h>
char *filename;
bool isFind=false;
char *string;
void *findString()
{
   if(!isFind)
   {
      char buffer[100];
      FILE *fp;
      fp=fopen(filename,"r");
      while(fscanf(fp,"%s",buffer)==1)
      {
         printf("%s  %s\n",string,buffer);
         if((strcmp(buffer,string)==0))
         {
            isFind=true;
            break;
         }
      }
   }
   pthread_exit(NULL);
}
```

```
int main(int argc, char *argv[]) {
   if (argc != 4) {
      printf("Usage: %s <filename> <integer> <string>\n", argv[0]);
      return 1;
   }

   filename = argv[1];
   int number = atoi(argv[2]); // Convert the second argument to an integer
   string = argv[3];

   printf("Filename: %s\n", filename);
   printf("Integer: %d\n", number);
   printf("String: %s\n", string);

   pthread_t *arrThreads=(pthread_t*)malloc(number*(sizeof(pthread_t)));

   for(int i=0;i<number;i++)
   {
      pthread_create(&arrThreads[i],NULL,&findString,NULL);
      sleep(2);
   }



   for(int i=0;i<number;i++)
   {
      pthread_join(arrThreads[i],NULL);
   }

   if(isFind) printf("Word find in the file.");
   else printf("Word is not find in the file.");
   return 0;
}
```

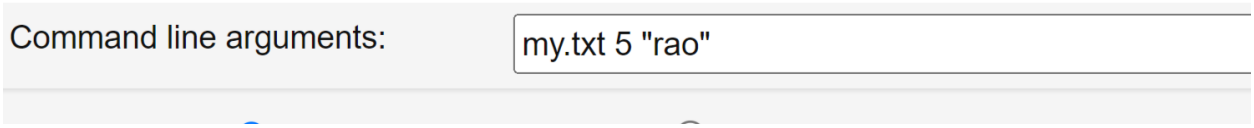
Screen shot of txt file:



Screen shot of outputs with sleep function:




```
Filename: my.txt
Integer: 5
String: waleed
waleed  Hi
waleed  my
waleed  name
waleed  is
waleed  rao
waleed  waleed
Word find in the file.

...Program finished with exit code 0
Press ENTER to exit console.
```



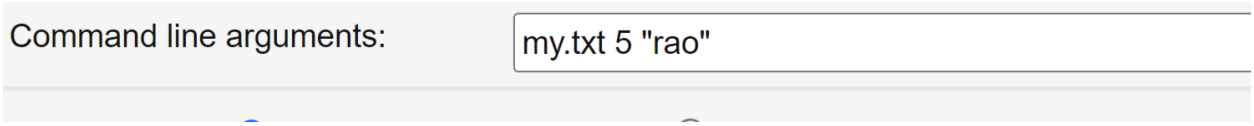

```
Filename: my.txt
Integer: 5
String: rao
rao  Hi
rao  my
rao  name
rao  is
rao  rao
Word find in the file.
```

It stops working where it finds the word and threads work one by one, so, ultimately it got the word in the first thread else won't find in any thread because the word Is not in the file.

Outputs with no sleep function multiple treads running at same time:

The outputs changes every time because the processor don't run equally every time.

Here is another example :



As you can see in this after rao(marked with yellow) there is no waleed because multiple threads working at same time and the reason of two rao in this output is that when the first rao(marked red) encountered the next get in through by any other thread at same time due to multithreading. But the function is completed by the rao(marked red).

Output of invalid word which is not in the file:

Run every thread but got nothing because word is not available.