PROGRAM 1:

```
//Toni Hunter 187009925
//ASSIGNMENT 5 PROGRAM 1
//
// A) Print the number of digits, alphabets, upper case alphabets, spaces, punctuations and graphs.
//-----------------------------------------------------------------------------------------------
// B) Convert all the upper case characters to lower case and convert all lower case to upper case and write to a separate file called C_Language_Convert.txt.
//
#include <stdio.h>
#include <string.h>
#include <ctype.h>

int main(void)
{
   char ch; //stores characters
   int digCount = 0;
   int AlCount = 0;
   int UpAlCount = 0;
   int spcCount = 0;
   int puncCount = 0;
   int LoCount = 0;
   int grphCount = 0;

   FILE *cfptr; //file handle
   FILE *CONVptr; // file handle for conversion
   cfptr = fopen("C_Language.txt", "r"); // open file for reading
   CONVptr = fopen("C_Language_Convert", "w"); // open conversion file for writing


   if (cfptr == NULL) // check file existance
   {
      printf("FNF!");
      return 0;
   }
   if (CONVptr == NULL)
   {
      printf("FNF!");
      return 0;
   }

   while ((ch = fgetc(cfptr)) != EOF) // put while loop in beginning, preference
      {
```

```
            if (islower(ch))
            {
                ch = toupper(ch);
            }

            else if (isupper(ch))
            {
                ch = tolower(ch);
            }
            fputc(ch, CONVptr);
        }
        rewind(cfptr); //rewinds pointer in cfptr since at the end of this while loop cfptr is EOF, other
count would be 0f


    //ch gets pointer to file and continues through each letter until EOF
    for (ch = getc(cfptr); ch != EOF; ch = getc(cfptr))
    {
        if (isalpha(ch)) // alphabet
        {
            AlCount++;
        }
        if(isspace(ch)) //space
        {
            spcCount++;
        }
        if (isupper(ch)) //uppercase
        {
            UpAlCount++;
        }
        if (islower(ch)) //lowercase
        {
            LoCount++;
        }
        if (isdigit(ch)) //digit
        {
            digCount++;
        }
        if (ispunct(ch)) //punctuation
        {
            puncCount++;
        }
        if (isgraph(ch)) //graph
        {
```

```
            grphCount++;
        }
    }
    fclose(CONVptr);
    fclose(cfptr); //always close at end


    printf("Printed below are the number of each item occurring in C_Language.txt. UC = upper
case, LC = lower case.\n");
    printf("\n");
    printf("Alphabets:            %d\n", AlCount);
    printf("UC Alphabets:         %d\n", UpAlCount);
    printf("LC Alphabets:         %d\n", LoCount);
    printf("Digits:              %d\n", digCount);
    printf("Graphs:               %d\n", grphCount);
    printf("SpacesS:              %d\n", spcCount);
    printf("Punctuations:         %d\n", puncCount);

    return 0;
}
```

PROGRAM 2:

```
//Toni Hunter 187009925
//ASSIGNMENT 5 PROGRAM 2
//
// Write a code to count the number of words (any token separated by space) from the file.
//
#include <stdio.h>
#include <stdlib.h>

int main(void)
{
    char data[5000];
    int WCount = 0;
    FILE *cfptr;
    cfptr = fopen("C_Language.txt", "r");

   if(cfptr == NULL) // check for file existience
   {
     printf("FNF!");
   }

   else
   {
      while (!feof(cfptr)) // not EOF
      {
            fgets(data, 5000, cfptr); // reads line from cfptr and stores in data
            char *tokenptr = strtok(data, "\n "); // assigns tokenptr to first token in data seprated by
\n and space

            while (tokenptr != NULL) // continue tokenizing until NULL
            {
              WCount++;
              tokenptr = strtok(NULL, "\n "); //get next token
            }
      }
      printf("The number of words present in the file: %d\n", WCount);
   }

}
```