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

/* Comment */
/*
To compile just cd into the CS433 folder.
Cd into the CS433 folder again B/c there is another folder named CS433 inside the CS433
 folder.
gcc -o HW1 HW_1.c to compile
./HW1 it was the < tintTale.txt to run the program
Nice finished!
*/

/* Struc */
struct keyword
{
    char *keyword;
    int counter;
};

/* Contains find_Str */
int find_str(char const *str, char const *substr)
{
    char *pos = strstr(str, substr);
    if (pos)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}

/* Contains getting the arguments and getline */
int main(int argc, char *argv[])
{
    /* These are for the command line */
    int keyword;
    int counter;
    /* These are for the getline */
    char *line = NULL;
    size_t maxlen = 0;
    ssize_t n;
    /* struct */
    /* malloc() */
    struct keyword list[argc + 1];
    int t;
    for (t = 0; t < argc - 1; t++)
    {
        list[t].keyword = argv[t + 1];
        list[t].counter = 0;
    }
    /* NEW WAY */
    while ((n = getline(&line, &maxlen, stdin)) > 0)
    {
        char *token = strtok(line, " ");
        while (token != NULL)
        {
            int i;
            for (i = 0; i < argc - 1; i++)
            {
                if (strcmp(token, list[i].keyword) == 0)
                {
                    list[i].counter = ++list[i].counter;
                }
                token = strtok(NULL, " ");
            }
        }
    }
    int i;
    for (i = 0; i < argc; i++)
    {
        printf("%s: %d\n", list[i].keyword, list[i].counter);
    }

    /* OLD CODE */

    /*
    if (argc == 1)
    {
        printf("There are no extra Command Line Arguments. \n");
    }
    if (argc >= 2)
    {
        for (keyword = 1; keyword < argc; keyword++)
        {
            while ((n = getline(&line, &maxlen, stdin)) > 0)
            {
                char *token = strtok(line, " ");
                while (token)
                {
                    if (find_str(token, argv[keyword]) == 1)
                    {
                        counter++;
                    }
                    token = strtok(NULL, " ");
                }
            }
            printf("%s: %d \n", argv[keyword], counter);
            counter = 0;
            free(line);
            line = NULL;
            n = 0;
            maxlen = 0;
        }
    }
    */
    return 0;
}
```