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

struct keyCount {
        char* keyword;
        int count;
};

int countKeyword(char* line, int n, char* keyword){
    // Lexical Analysis
    // Chains character after character together to recognize words in a string.
    // NOTE: Punctuation does not need to be handled special in this case,
    //        as it recognizes the keyword before adding punctuation.
    //  Ex: "it," gets recognized as "it" before adding & rejecting "it,", so the
    //        keyword is still counted regardless of punctuation.
    //
    // Input: String pointer, char* line, for the string to be analyzed
    //        Integer, int n, representing line’s length
    //        String Pointer, char* keyword, for the keyword to search for
    // Output: Integer that equals the amount of instances of the given keyword.
    int count = 0;
    size_t maxlen = sizeof(char)*n;
    int j;
    char* string = malloc(maxlen);
    for (j=0; j<n; j++){
        // printf("%c\n", line[j]);
        if(line[j] == ’ ’ || line[j] == ’\0’){
            // NOTE: We check this string *after* encoutnering whitespace,
            //   as keywords starting with the keyword we are looking for could
            //   be a false positive.

            // If whitespace encountered, check then reset string
            if(strcmp(string,keyword) == 0){
                //printf("--Got count!\n");
                //printf("%s, %s\n",string,keyword);
                count++;
            }
            free(string);
            string = malloc(maxlen);
        }
        else{
            // char encountered is part of word
            char temp[] = {line[j],’\0’};
            strcat(string,temp);
        }
    }
    free(string);
    return count;
}

void main(int argc, char** argv){
    char *line=NULL;
    size_t maxlen=0;
    ssize_t n;
    int i;

    struct keyCount counts[argc-1];

    for(i = 1;i < argc; i++){
        struct keyCount new;
        new.keyword = argv[i];
        new.count = 0;
        counts[i] = new;
    }

    while ((n = getline(&line, &maxlen, stdin)) > 0){
        int i;
        for(i = 1;i < argc; i++){
            // printf("%s\n",argv[i]);
            counts[i].count += countKeyword(line,n,counts[i].keyword);
        }

    }
    int j;
    int longest_keylen = 0;
    for(j = 1; j < argc; j++){
            if(strlen(counts[j].keyword) > longest_keylen){
                longest_keylen = strlen(counts[j].keyword);
            }
    }
    for(i = 1;i < argc; i++){
        int standar = 7;
        printf("# of %s’s",counts[i].keyword);
        int space = longest_keylen-strlen(counts[i].keyword);
        for(j=0;j<space;j++){
            printf(" ");
        }
        printf("  :  %i\n", counts[i].count);
    }
}
```