```c
/*
*C program thaat a variable number of keywords as command line argument,
*reads text from the standard input stream, searches the keywords in
*the input stream and shows how many times it is present.
*@version
*@author Revanth Kari <rkari@uab.edu>
*/

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

//defining the structure with 2 entries
typedef struct result_table{
char *keyword;
int count;
}table;

//Read the keywords provided as the command line argument
int main(int argc, char **argv){
int i;

/*Dynamic memory allocation of the Array*/
table *result = malloc(argc * sizeof *result);

//Initializing the keywords and the count
for(i = 0; i < argc-1; i++){
result[i].keyword = argv[i+1];
result[i].count = 0;
}

//Use getline() to read the given text from command line
char *line=NULL;
size_t maxlen=0;
ssize_t n;
char *token;
while ((n = getline(&line, &maxlen, stdin))>0) {

/*strtok: A library function breaks the string into
a serires of tokens using delimiter.*/

token = strtok(line," ");
while(token != NULL){

// printf("%s\n", token ); //To print the Token
for(i=0; i < argc-1; i++){

//strcspn:Function to remove the ’\n’ from the tokens.
token[strcspn(token, "\n")] = 0;

//strcmp:Function to compare the keywords and token.
if(strcmp(token, result[i].keyword)== 0){
result[i].count++;  //updating the keyword count.
}

}

token = strtok(NULL, " ");
}

}

// Printing the Array of keyword and count.
for (i = 0; i < argc-1; i++) {
printf("%s =", result[i].keyword);
printf("%d \n", result[i].count);
}

//Library function to deallocate the memory.
free(result);
free(line);

return 0;

}
```