

UMass Boston CS 240  
Homework 2  
Due 6/14/2019 11:59 pm

## 1 Histogram of Word Lengths

Write a C program that reads from `stdin` till EOF and analyzes the lengths of the words in the input. Let's consider all alphanumeric characters as words, and all non-alphanumeric characters as delimiters. For example, each of the following is a word.

- Homework
- CS240
- 2

The following strings should be broken into multiple words.

- “you’ll” has two words: “you” and “ll”.
- “ALL’S” has two words: “ALL” and “S”.
- “UTF-8” has two words: “UTF” and “8”.
- “2/19/2019 17:00” has five words: “2”, “19”, “2019”, “17”, and “00”.
- “www.gutenberg.org” has three words: “www”, “gutenberg”, and “org”.

You can build upon the word counting code on page 20 of K&R. As you read a word one character at a time, keep track the number of characters you have read. When you reach the end of a word, you have its length. Then you increment a counter that keeps track the number of words of this particular length. Use an array of these counters. I will test your code with `CompleteShakespeare.txt`. The longest word you will encounter has 27 characters.

For output, you should print 27 lines. In each line, you print the length (width 2), a space, the number of words of that length (width 6), a space, and several asterisks. Use one asterisk for each 4,000 words. If there are fewer than 4,000, you still print one asterisk for them, because we cannot print a fractional asterisk. For example, print one asterisk for 1 to 4,000 words, and two asterisks for 4,001 to 8,000 words, and so on. The asterisks constitute the histogram of word lengths. Histograms are usually printed vertically, but here it is printed horizontally because this is easier. On `CompleteShakespeare.txt`, your code should print exactly like Figure 1.

```

1  63691 *****
2 166375 *****
3 204211 *****
4 223161 *****
5 121472 *****
6  80386 *****
7  59379 *****
8  35083 *****
9  20351 *****
10 10067 ***
11  3771 *
12  1353 *
13   454 *
14   247 *
15    77 *
16     3 *
17     4 *
18     0
19     0
20     0
21     0
22     0
23     0
24     0
25     0
26     0
27     1 *

```

Figure 1: Output for CompleteShakespeare.txt

## 2 Lastly

Compile and run as follows:

```
user80@itserver:~/$gcc -Wall histo.c -o histo
user80@itserver:~/$./histo < CompleteShakespeare.txt
```

Write plenty of comments to explain your code – how you determine a character is alphanumeric, how you extract one word, and how you convert a count to the number of asterisks to print.

Write a **Report.txt** that discusses what you found difficult about this assignment, how you planned your approach to it, and what you learned completing it.

Send me only:

1. histo.c
2. Report.txt