

C Assignment - Histogram

Swapnil Basak
EE11B122
IIT Madras

August 13, 2012

Problem Statement

The task is to create a C program that goes through a text file and prints the statistics of number of words with different lengths. We will consider words of length of 3 to 10 for our statistics. This program is written by indexing through strings using arrays.

Code

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 #define MAXLENGTH 512
4 /*
5  Program to find out no. of words with N
6  chars breaking at ' ','.',',','/t','/0',' '
7  This is the Array method implementation.
8  Swapnil Basak
9  EE11B122
10 */
11
12 //Histogram
13 void histogram(int p[10])
14 {
15     int i=0;
16     int max=0;
17     int j=1;
18     // Find max occurrences for upper limit of Y-axis
19     for(i=0;i<9;i++)
20     {
21         if(p[i]>max)
22         {
23             max=p[i];
24         }
25     }
26     printf("\n");
27     // Plotting Logic
28     for(i=max;i>0;i--)
29     {
30         printf("|",i);
31         for(j=1;j<11;j++)
32         {
33             if(p[j]>=i)
34             {
35                 printf("# ");
36             }
37             else
38             {
39                 printf(" ");
40             }
41         }
42     }
```

```

41     }
42     printf("\n");
43 }
44 printf("|_____ \n");
45 printf(" |1 2 3 4 5 6 7 8 9 10");
46 }
47
48 int main(int argc, char **argv)
49 {
50     /* Expects a filename too, check for that */
51     if(argc != 2)
52     {
53         printf("Usage ./a.out <filename>");
54         exit(1);
55     }
56     /* File pointer */
57     FILE *fp;
58     fp=fopen(argv[1], "r");
59     if(fp == NULL)
60     {
61         printf("File could not be opened");
62         exit(2);
63     }
64     char buf[MAXLENGTH];
65     int counter = 0;
66     char wordcount[11];
67     int wordlen=0;
68     int tempcount=0;
69     /* Initialize all to 0 */
70     for(counter=0;counter<11;counter++)
71     {
72         wordcount[counter]=0;
73     }
74     /* Loop till EOF */
75     while(fgets(buf, MAXLENGTH, fp))
76     {
77         printf("Parsing text line --> %s", buf);
78         for(counter=0;counter<MAXLENGTH;counter++)
79         {
80             /* Split at specified characters and check if it is in between a sentence
81             or at the beginning */
82             if(buf[counter]==' ' || buf[counter]=='\t' || buf[counter]==' , ' )
83             {
84                 /* Second condition avoids double spaces and spaces after periods */
85                 if(counter!=0)/* &&((counter-tempcount)>1))*/
86                 {
87                     wordlen=counter-tempcount;
88                     wordcount[wordlen]+=1;
89                     wordlen=0;
90                 }
91                 /* Move tempcount to present space+1 */
92                 tempcount=counter+1;
93             }
94             /* Specific logic for EOF as vars need to be reset */
95             if(buf[counter]=='\0')
96             {
97                 wordlen=counter-tempcount-1;
98                 wordcount[wordlen]+=1;
99                 wordlen=0;
100                tempcount=0;
101                /* End iteration */
102                break;
103            }
104        }

```

```

105 }
106 for(counter=1;counter<11;counter++)
107 {
108     printf(" \nThe no. of %d lettered words are %d \n", counter, wordcount[counter]);
109 }
110 histogram(wordcount);
111 fclose(fp);
112 return 1;
113 }

```

Output

```

1 ./a testfile.txt
2 Parsing text --> This is a test file with default words to see if it the histogram draws corre
3
4 The no. of 1 lettered words are 1
5 The no. of 2 lettered words are 4
6 The no. of 3 lettered words are 2
7 The no. of 4 lettered words are 4
8 The no. of 5 lettered words are 2
9 The no. of 6 lettered words are 0
10 The no. of 7 lettered words are 1
11 The no. of 8 lettered words are 0
12 The no. of 9 lettered words are 2
13 The no. of 10 lettered words are 0
14
15 |  *   *
16 |  *   *
17 |  * * * *      *
18 |* * * * *      * *
19 |_____
20 |1 2 3 4 5 6 7 8 9 10

```

Algorithm

An array index ptr is declared that indexes throughout the buffer string. Two flags are used for the logic.

- A file stream is opened with specified txt file, exception is raised if not
- *buf* is initialized per line with MAXLENGTH till EOF is reached
- *counter* that travels with the array index
- *tempcount* that relates to the last position of *counter*
- *wordcount* is stored in an array with *wordcount[n]* denotes no. of words consisting of n letters
- wordcount is passed to histogram() and proccessed
- The max element is found in the array. This serves as the maximum Y axis limit.
- Consequently, the text graph is filled with a double for-loop if for *wordcount[n]* the max element at that iteration is smaller than max(*wordcount[n]*)
- Finally results are displayed