

Learning summary:

Study: 5 Hours

Exercises: 2.5 Hours

Day 15

Exercise 1-13:

Question:

Write a program to print a histogram of the lengths of words in its input. It is easy to draw the histogram with the bars horizontal; a vertical orientation is more challenging.

Solution:

Source Code:

```
#include <stdio.h>
#define MAXSIZE 1000

int freq_of_individual_wordsize[MAXSIZE+1];
int max_window_height = 0;
void draw_histogram(){
    int histogram_bar_height, i, j;
    for(histogram_bar_height = max_window_height; histogram_bar_height > 0;
    histogram_bar_height--){
        printf("%3d|", histogram_bar_height);
        for(j = 0; j < MAXSIZE; j++){
            if(freq_of_individual_wordsize[j] == 0) continue;
            if(freq_of_individual_wordsize[j] < histogram_bar_height) {
                printf(" ");
            }
            else {
                printf(" *");
            }
        }
        printf("\n");
    }
    for(i = 0; i <= MAXSIZE; i++) {
        if(i == 0){
            printf(" ");
        }
        else if(freq_of_individual_wordsize[i]) {
            printf("---");
        }
    }
    printf("\n");

    for(i = 0; i <= MAXSIZE; i++) {
        if(i == 0){
```

```

        printf(" ");
    }
    else if(freq_of_individual_wordsize[i]) {
        printf("%3d", i);
    }
}
printf("\n");
}

void process_input() {
    int each_string_length = 0;
    char ch;
    while((ch = getchar()) != EOF) {
        if(ch == ' ' || ch == '\t' || ch == '\n') {
            if(each_string_length) {
                freq_of_individual_wordsize[each_string_length]++;
                if(freq_of_individual_wordsize[each_string_length] > max_window_height) {
                    max_window_height = freq_of_individual_wordsize[each_string_length];
                }
            }
            each_string_length = 0;
        }
        else {
            each_string_length++;
        }
    }
}

```

Functions:

process_input() function :

The word_length holds length of word of different sizes that is used to update the frequency of word of different sizes. This condition `if(ch == ' ' || ch == '\t' || ch == '\n')` checks if there is any space tabs or newline in input. It determines that if the condition is true then we getting an individual word. Because a space or tab or newline separates one word to the next word. This condition `if(word_length)` then checks if there is any individual word in input, if it is true then frequency of the word having same size is incremented in word_length'th indexes of the frequency[] array. Then this condition `if(frequency[word_length] > max_window_height)` checks if the maximum frequency of words having same sizes (frequency[word_length]) is bigger than the window size. If bigger then updates the max window height with the maximum frequency of words having same sizes. To adjust the maximum window height according to the highest frequency the "max_window_height" is updated with the maximum frequency of the word having same sizes. After that word_length is set to 0 if there is no word. Again if we reached at the end of any individual word which is checked by this condition `{ if(ch == ' ' || ch == '\t' || ch == '\n') }` above then the wordlength is incremented by one.

draw_histogram()function :

Frequency character array stores the frequency of the occurrences of individual words that have same sizes in the input stream. And the size of the array is 1000+1 which means the maximum frequency can be 1001. 1 is added with 1000 to ignore 0 and start from 1 as the frequency of anything is always positive. The function draw_histogram() prints the histogram in output console. The 0'th index is skipped by this "condition if(frequency[j] == 0) continue;" because, frequency of anything is always > 0. Then the spaces above the histogram bars is printed to print the height of bar according to its frequencies by checking this condition "if(frequency[j] < histogram_bar_height)".

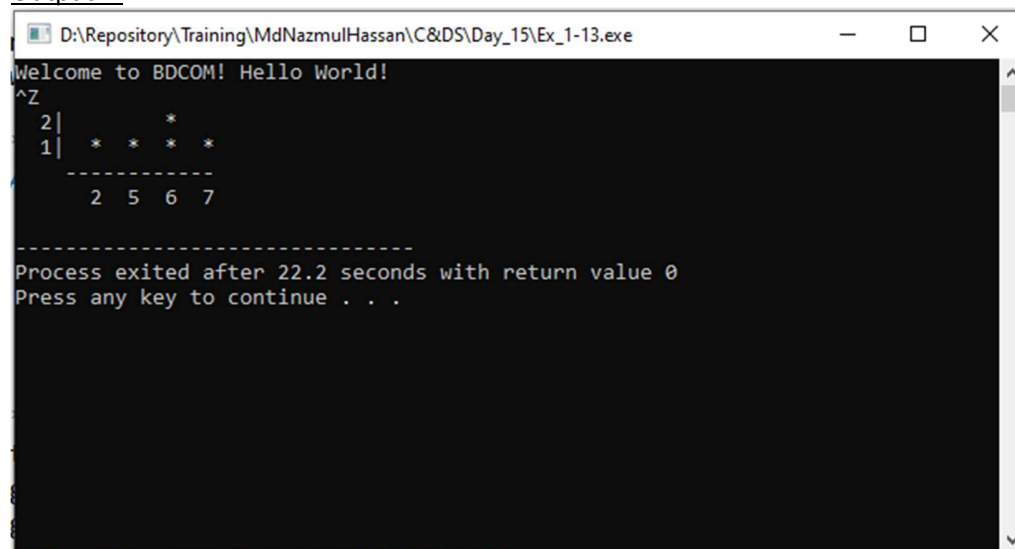
main():

The main() function invokes the **process_input()** and **draw_histogram()** functions and the histogram is printed according to the input string.

Input & Output:

Input 1: Welcome to BDCOM! Hello World!

Output 1:



```
D:\Repository\Training\MdNazmulHassan\C&DS\Day_15\Ex_1-13.exe
Welcome to BDCOM! Hello World!
^Z
 2|      *
 1|  *  *  *  *
  ---
    2  5  6  7

-----
Process exited after 22.2 seconds with return value 0
Press any key to continue . . .
```

Input 2:

welcome welcome welcome to to to BDCOM BDCOM BDCOM !!!

Output 2:

```
D:\Repository\Training\MdNazmulHassan\C&DS\Day_15\Ex_1-13.exe
welcome welcome welcome to to to BDCOM BDCOM BDCOM !!!
^Z
 3| * * * *
 2| * * * *
 1| * * * *
-----
    1 2 5 7

-----
Process exited after 96.03 seconds with return value 0
Press any key to continue . . .
```

Input 3:

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Output 3:

```
D:\Repository\Training\MdNazmulHassan\C&DS\Day_15\Ex_1-13.exe
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text
ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survive
d not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in t
he 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software l
ike Aldus PageMaker including versions of Lorem Ipsum.
^Z
18|      *
17|      *
16|      *
15|      *
14|    * *
13|    * * *
12|    * * *
11|    * * *
10|    * * *
 9|    * * *
 8|    * * *      *
 7|    * * *      *
 6|    * * *      * *
 5|    * * *      * *
 4|    * * *      * *
 3|    * * *      * *
 2|    * * *      * *
 1|    * * *      * *
-----
    1 2 3 4 5 6 7 8 9 10 11 12

-----
Process exited after 52.26 seconds with return value 0
Press any key to continue . . .
```

Exercise 1-14:

Question:

Write a program to print a histogram of the lengths of words in its input. It is easy to draw the histogram with the bars horizontal; a vertical orientation is more challenging.

Solution:

Source Code:

```
#include <stdio.h>
#define MAXSIZE 256

int freq_of_individual_character[MAXSIZE+1];
int max_window_height = 0;

void draw_histogram(){
    int height, i, j;
    for(height = max_window_height; height > 0; height--){
        printf("%3d |", height);
        for(j = 0; j < MAXSIZE; j++){
            if(freq_of_individual_character[j] == 0) continue;
            if(freq_of_individual_character[j] < height) {
                printf(" ");
            }
            else {
                printf(" *");
            }
        }
        printf("\n");
    }
    for(i = 0; i <= MAXSIZE; i++) {
        if(i == 0){
            printf(" ");
        }
        else if(freq_of_individual_character[i]) {
            printf("---");
        }
    }
    printf("\n");

    printf(" ");
    for(i = 0; i < MAXSIZE; i++) {
        if(freq_of_individual_character[i]){
            if(i == ' '){
                printf("\' \'");
            }
            else if(i == '\n') {
                printf("\n");
            }
            else if(i == '\t'){
```

```

        printf(" \\t");
    }
    else if(freq_of_individual_character[i]) {
        printf("%3c", i);
    }
}
printf("\n");
return;
}

void process_input() {
    char ch;
    while((ch = getchar()) != EOF) {
        freq_of_individual_character[ch]++;
        if(freq_of_individual_character[ch] > max_window_height) {
            max_window_height = freq_of_individual_character[ch];
        }
    }
    return;
}

int main() {
    process_input();
    draw_histogram();
    return 0;
}

```

Functions:

process_input() function :

The word_length is used to update the frequency of individual character in freq_of_individual_character[] array. This condition “while((ch = getchar()) != EOF)” then checks if there is any individual character in input, if it is true then frequency of the individual character incremented in word_length'th indexes of the frequency[] array. Then this condition “if(freq_of_individual_character[ch] > max_window_height)” checks if the maximum frequency of words having same sizes (frequency[word_length]) is bigger than the maximum window size, if it is bigger then updates the max window height with the maximum frequency of words having same sizes. To adjust the maximum window height according to the highest frequency, the “max_window_height” is updated with the maximum frequency of among all characters.

draw_histogram() function :

Frequency character array stores the frequency of the occurrences of individual character in the input stream. And the size of the array is 256+1 which means the maximum frequency can be 257. 1 is added with 256 to ignore 0 and start from 1 as the frequency of anything is always positive.

The function draw_histogram() prints the histogram in output console. The 0'th index is skipped by this “condition if(frequency[j] == 0) continue;” because, frequency of anything is always > 0. Then the spaces above the histogram bars is printed to print the height of bar according to its frequencies by checking this condition “if(frequency[j] < histogram_bar_height)”.

main():

The main() function invokes the **process_input()** and **draw_histogram()** functions and the histogram is printed of the frequencies of different characters in its input.

Input & Output:

Input 1:

Aabbccddeeffggghhhijklllllmmmmmmmmoooooooooooooppqqqqqqrrrrrrsssstttt

Output 1:



Input 2:

Welcome to BDCOM! Hello World! How are you? \$

Output2:

```
D:\Repository\Training\MdNazmulHassan\C&DS\Day_15\Ex_1-14.exe
Welcome to BDCOM! Hello World! How are you? $
^Z
 8| *
 7| *
 6| *
 5| *
 4| *
 3| *
 2| *
 1| *
-----
\n' ' ! $ ? B C D H M O W a c d e l m o r t u w y
-----
Process exited after 3.988 seconds with return value 0
Press any key to continue . . .
```

Input 3:

Asia ^ Pacific \$ Support & Center... BDCOM \n Dhaka / BangLaDesh

Output 3:

```
D:\Repository\Training\MdNazmulHassan\C&DS\Day_15\Ex_1-14.exe
Asia ^ Pacific $ Support & Center... BDCOM \n Dhaka / BangLaDesh
^Z
10| *
 9| *
 8| *
 7| *
 6| *
 5| *
 4| *
 3| *
 2| *
 1| *
-----
\t \n' ' $ & . / A B C D L M O P S \ ^ a c e f g h i k n o p r s t u
-----
Process exited after 57.5 seconds with return value 0
Press any key to continue . . .
```