Playing With Strings Assignment

Juan Velasquez

Code:

/\*

Student Name: Juan Velasquez

Date Written: 9/6/2021

Purpose of program: This program manipulates user-inputted string in a bunch

of ways like showing it normally, reversed, vertical and finally in a triangle.

The program also returns the length of the string to the user and asks the user

if they would like to enter another string or to quit the program.

\*/

#include <stdio.h>

//Function Declarations

int getLen(char str[]);

void forward(char str[]);

void backward(char str[], int len);

void vertical(char str[], int len);

void triangle(char str[], int len);

void printLen(int len);

char getUserChoice();

/\*

This function gets the length of the string by utilizing a for loop. The for loop

goes through the string and will continue to iterate by adding 1 to the out variable

through it until the current character at index i is equal to '\0' or the end of

the string. It then returns the total length of the string.

\*/

int getLen(char str[]){

int out = 0;

for(int i = 0; str[i]!='\0'; i++){

out++;

}

return out;

}

/\*

This function just prints the string out normally so nothing complex is needed.

\*/

void forward(char str[]){

printf\_s("\n\nForward: \n");

printf\_s("\n%s", str);

}

/\*

This function prints the string backwards. In order to do that, we need length of the

string passed through. We then run a for loop where i is initialized to the length and

we run until i is greater than or equal to 0 while subtracting 1 from i. Basically,

we iterate through the character array (String) backwards and print each character

on each loop.

\*/

void backward(char str[], int len){

printf\_s("\n\nBackward: \n");

printf\_s("\n");

for(int i = len; i >= 0; i--){

printf\_s("%c", str[i]);

}

}

/\*

This function prints the string vertically. In order to do that, we need length of the

string passed through. We then run a normal for loop where we print a newline with the

character at i until the for loop reaches length.

\*/

void vertical(char str[], int len){

printf\_s("\n\nVertical: \n");

for(int i = 0; i < len; i++){

printf\_s("\n%c", str[i]);

}

}

/\*

This function prints the string in a triangle. In order to do that, we need length of the

string passed through. This function is a little more complex so I will explain it line by

line.

\*/

void triangle(char str[], int len){

printf\_s("\n\nTriangle: \n");

/\*

Variables needed to calculate things about the triangle like how many rows and the

variables used to calculate that.

\*/

int rows = 2;

int total = 1;

int next = 2;

/\*

While true, we are going to add a row while total + next are less than the total length

of the string. For example, if we have a string of length 10, the values would look like:

Total | Next | Sum | Rows

1 | 2 | 3 | 2

3 | 3 | 6 | 3

6 | 4 | 10 | 4

which is correct because:

1

2 3

4 5 6

7 8 9 0

\*/

while(1){

total = total + next;

if(total + next < len){

rows++;

next++;

}

else{

break;

}

}

/\*

We then calculate the remainder from the string length and total which will be used

for the shape.

\*/

int remainder = len - total;

/\*

An integer array is then created to store the values of how many elements will be in

each row.

\*/

int cols[rows];

/\*

This for loop will fill out the columns by adding one to i and setting that equal to

cols[i]. I.e.: cols[0] = 1, cols[1] = 2, cols[2] = 3, and so on.

\*/

for(int i = 0; i < rows; i++){

cols[i] = i+1;

}

/\*

Then we use a for loop to iterate through the remainder to add one to each row

starting from the bottom. I.e.: remainder = 2, cols[0] = 1, cols[1] = 3, cols[2] = 4

\*/

for(int i = 0; i < remainder; i++){

cols[rows-i-1]++;

}

//Total needs to be set equal to 0 for the printing later.

total = 0;

/\*

Calculate the size of each row by getting the last cols[] value and multiplying it by

2 and adding 1 to it.

\*/

int rowLen = (cols[rows-1]\*2)+1;

/\*

This for loop will iterate for each row, printing a new line on each iteration.

\*/

for(int i = 0; i < rows; i++){

printf\_s("\n ");

/\*

This for loop will print out the empty spaces before the characters according

to this formula: (rowLen - (cols[i]\*2)-1)/2

I.e.: rowLen = 7, cols[i] = 1; (7-((1\*2)-1)/2 = (7-1)/2 = 3 so three spaces before

the first character of that line.

\*/

for(int j = 0; j < ((rowLen - (cols[i]\*2)-1)/2); j++){

printf\_s(" ");

}

/\*

This prints the character at index total plus an additional space and then adds one

to the total counter.

\*/

for(int j = 0; j < cols[i]; j++){

printf\_s("%c ", str[total]);

total++;

}

}

}

/\*

This function prints the length of the string to the user.

\*/

void printLen(int len){

printf\_s("\n\nThe string is %d characters long.\n", len);

}

/\*

This function gets the users input for the menu at the end of every loop. I use toupper to

make all lowercase inputs into uppercase automatically.

\*/

char getUserChoice(){

printf\_s("\nWould you like to enter another string?");

printf\_s("\nEnter Y/y if you do or enter Q/q if you want to quit.");

char temp;

scanf("%c", &temp);

return toupper(temp);

}

//The main function that the program runs.

int main(void){

//Variable to store the user input.

char input[100];

int length = 0;

//Print out asking what string the user wants.

printf\_s("\nPlease enter the string that you want to use (No longer than 100 letters): ");

//This is used to track the user input through the menu.

char selection;

/\*

Run one loop of string manipulation before going into the menu as the menu is presented

after one loop.

\*/

scanf("%s", input);

length = getLen(input);

forward(input);

backward(input, length);

vertical(input, length);

triangle(input, length);

printLen(length);

//Menu Design according to menu sample code.

do{

selection = getUserChoice();

switch(selection){

case 'Y':

printf\_s("\nPlease enter the string that you want to use (No longer than 100 letters): ");

scanf("%s", input);

length = getLen(input);

forward(input);

backward(input, length);

vertical(input, length);

triangle(input, length);

printLen(length);

break;

default:

system("pause");

break;

}

}

while(selection != 'Q');

printf\_s("\nGoodbye!");

//Exit program.

return 0;

}

Screenshots:

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated