String Decoder Assignment

Juan Velasquez

Code:

/\*

Student Name: Juan Velasquez

Date Written: 9/10/2021

Purpose of program: This program encodes a string provided by

the user and prints the original string proceeded by the encoded

string. It then asks the user if they want to enter another

string.

\*/

#include <stdio.h>

//Function Declarations

int getLen(char str[]);

char encodeChar(char in);

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

char getUserChoice();

/\*

This function gets the length of a string and returns it as

an integer.

\*/

int getLen(char str[]){

int out = 0;

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

out++;

}

return out;

}

/\*

This is the primary function that prints out the original

string and then proceeds to encodes the string and print it

out character by character

\*/

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

//Print the original String

printf\_s("\nYou Entered : %s\nThe Encoded string is : ", str);

/\*

Loop through the string and encode each character and print

the result of each character.

\*/

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

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

}

}

/\*

This function checks if the character passed through matches

any character in the alphabet and returns what the encoded

character should be. The inputted character is also made

uppercase in beginning of the function just to make the

arguements in the if and else if statements cleaner.

\*/

char encodeChar(char in){

in = toupper(in);

if(in == 'A'){

return 'H';

}

else if(in == 'B'){

return 'I';

}

else if(in == 'C'){

return 'V';

}

else if(in == 'D'){

return 'J';

}

else if(in == 'E'){

return 'K';

}

else if(in == 'F'){

return 'W';

}

else if(in == 'G'){

return 'L';

}

else if(in == 'H'){

return 'A';

}

else if(in == 'I'){

return 'M';

}

else if(in == 'J'){

return 'Y';

}

else if(in == 'K'){

return 'N';

}

else if(in == 'L'){

return 'B';

}

else if(in == 'M'){

return 'X';

}

else if(in == 'N'){

return 'O';

}

else if(in == 'O'){

return 'Z';

}

else if(in == 'P'){

return 'C';

}

else if(in == 'Q'){

return 'T';

}

else if(in == 'R'){

return 'P';

}

else if(in == 'S'){

return 'U';

}

else if(in == 'T'){

return 'D';

}

else if(in == 'U'){

return 'Q';

}

else if(in == 'V'){

return 'S';

}

else if(in == 'W'){

return 'R';

}

else if(in == 'X'){

return 'E';

}

else if(in == 'Y'){

return 'G';

}

else if(in == 'Z'){

return 'F';

}

}

/\*

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);

}

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 option = 'z';

/\*

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);

encodeString(input, length);

//Menu Design according to menu sample code.

do{

option = getUserChoice();

switch(option){

case 'Y':

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

scanf("%s", input);

length = getLen(input);

encodeString(input, length);

break;

default:

system("pause");

break;

}

}

while(option != 'Q');

printf\_s("\nGoodbye!");

//Exit program.

return 0;

}

Screenshots:

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated