Sam Dinkelman

12/5/2018

CIS 247

Assignment 3 Lab Report

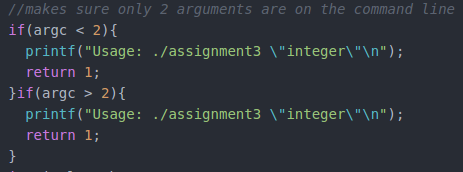
Introduction:

For this assignment we were tasked with writing a program to print numbers entered by the user in a seven segment display format.

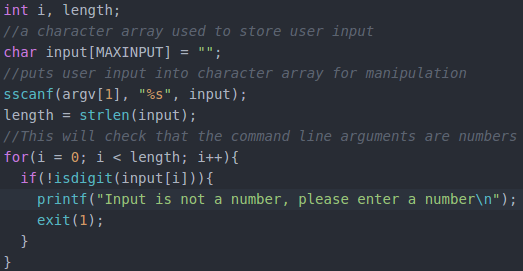
Process:

For this lab I began by reading the instructions thoroughly which explain that we need to split each numbers into 3 zones a top a middle and a bottom. So immediately I know that I will have 3 void functions one for each of the zones. The instructions also specifiy that we must check that the user entered a number with the program on the command line so I start here.

I first check that enough arguments are entered on the command line since there should be only 2 arguments, the program we run and the integer.

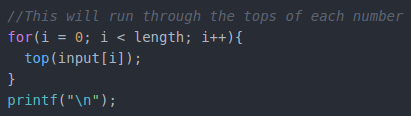


Next I create an index integer and a character array input and sscanf our second string entered on the command line into this character array input. Then we store the length of this into our integer length. Next up we run a for loop that loops us through each digit of the string of integers entered. Then we use a standard library function isdigit and state that if we find a character that isn’t a digit we will print an error message and exit the program.

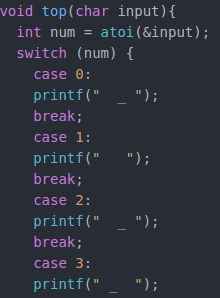


Up at the top of the program we define a max input our program will accept as well since printing more than 42 characters can overflow the screen size and mess up the formatting.

  
Next up I start another for loop which will pass each character of the input to my top function. After which we will print a new line, so it will go through printing all of the tops of the numbers and then start a new line for the middle.



Up in my Top function I take in the character and turn it into an integer with atoi. Then I take that integer and start a switch statement which goes through every number from 0-9. For each case I do a print statement for the correct formatting and break the switch statement after so that it will only print once for each number entered. So after it finds which case it needs it will break and go back down to main where the next number will be passed to the top function until they all have been.



I perform this same series of steps for both middle and bottom. So I won’t go into detail on them here since the only difference between them is the formatting of the specific numbers.

Testing:

To test my program I first enter something which is not a number to test what it will do:



So here when I enter aaaaaaa I get the error that input is not a number.

Next I tried doing a dashed number. It found that this was not a number since I didn’t implement the negative sign it will see the dash and print the error and exit.



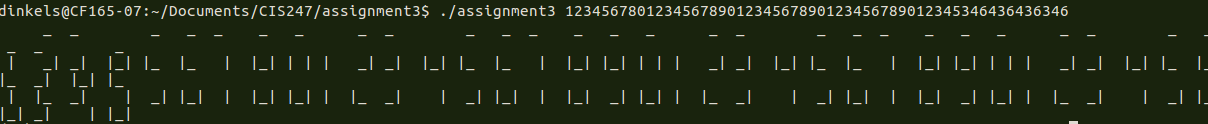
The same happens for all the symbols on the keyboard which I enter, even if it sees numbers if it gets a symbol it will print the error and exit.

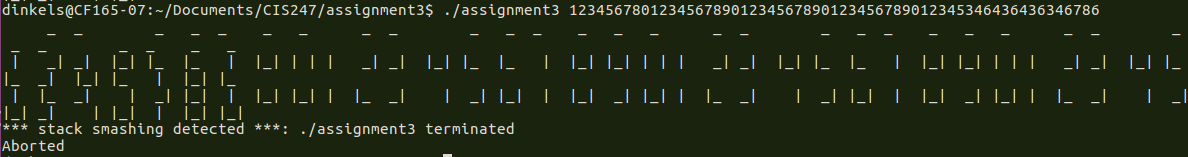
Next I wanted to try entering more than 2 arguments.



Here we can see that it prints the usage of the program and won’t run the rest of the program unless the specific input is entered.

Now I want to try and test if I have more than 42 numbers since that was the max input I set.



So here we can see that if a number larger than 42 is entered it does accept it but the formatting is all messed up. Additionally if we keep adding numbers up to 50 we get an error that stack smashing is detected. 

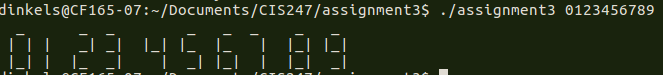
I’m going to try to fix this in my code since it won’t be too difficult to correct.

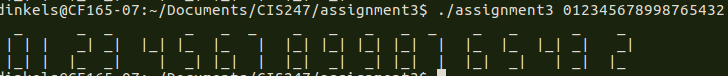


Now we can see that if you enter more than 42 numbers it will print an error message that You have entered too many numbers. No more stack smashing and no more incorrect formatting.

Results:

So here we can see that my program will print all the numbers from 0-9 and any combination





Also if you enter more than 42 numbers it will print this error statement.



This is to ensure that the formatting stays correct since if you enter too many numbers it will mess up the formatting.

Also if you enter less or more than two arguments on the command line when you run the program it will print an error statement.





Conclusions:

Overall this assignment was not very challenging and I felt as though it didn’t really map to what we were learning in the course very much. This assignment felt really simple and basic and I feel as though we could have tackled this assignment in the first month of class. It was nice to have kind of a break from hard assignments and labs but I feel as though this one was a little basic for our progression in the course.

References and acknowledgments:

In this lab I referenced the K&R book for the syntax of a switch statement but that was the only reference I used.