Assignment 2 – Hangman Report

Caden Roberts

CSE 13S – Winter 24

Purpose

Audience for this section: Pretend that you are working in industry, and write this paragraph for your boss. You are answering the basic question, "What does this thing do?". This section can be short. A single paragraph is okay.

Do not just copy the assignment PDF to complete this section, use your own words.

ANS:

The purpose of this assignment it to implement a Hangman game along 4 functions that make up its logic. We will work with strings, arrays, char *'s, chars, and integers.

Questions

Please answer the following questions before you start coding. They will help guide you through the assignment. To make the grader's life easier, please do not remove the questions, and simply put your answers below the text of each question. To fill in the answers and edit this file, you can upload the provided zip file to overleaf by pressing [New project] and then [Upload project].

Guesses

One of the most common questions in the past has been the best way to keep track of which letters have already been guessed. To help you out with this, here are some questions (that you must answer) that may lead you to the best solution.

- How many valid single character guesses are there? What are they?

  ANS:

  There are 26 valid lowercase character guesses, a through z.

- Do we need to keep track of how many times something is guessed? Do you need to keep track of the order in which the user makes guesses?

  ANS:

  We don't need to keep track of how many times something is guessed, as we will just re-prompt the user for a different guess. We will only track, in alphabetical order, the eliminated characters that are not in the secret.

- What data type can we use to keep track of guesses? Keep in mind your answer to the previous questions. Make sure the data type you chose is easily printable in alphabetical order.

ANS:

We should use a 26 char string that only prints present letters to keep everything in alphabetical order.

- Based on your previous response, how can we check if a letter has already been guessed.

ANS:

We will utilize the string_contains_character() function we define by passing the character pointer for the displayed secret and the eliminated letters and checking if the character is already contained in either. [1]

Strings and characters

- Python has the functions `chr()` and `ord()`. Describe what these functions do. If you are not already familiar with these functions, do some research into them.

ANS:

The chr funtion will return a character to an integer input and the ord function will return an integer to a character input.

- Below is some python code. Finish the C code below so it has the same effect. [2]

```
x = 'q'
print(ord(x))
```

ANS:

C Code:

```
char x = 'q';
\textbf{printf("%i", x-'0');}
```

- Without using `ctype.h` or **any** numeric values, write C code for `is_uppercase_letter()`. It should return false if the parameter is not uppercase, and true if it is.

ANS:

```
#include <stdbool.h>
char is_uppercase_letter(char x){
    (x >= 'A' && x <= 'Z');
}
```

- What is a string in C? Based on that definition, give an example of something that you would assume is a string that C will not allow.

ANS:

A string is defined as an array of characters terminated by a null character.

- What does it mean for a string to be null terminated? Are strings null terminated by default?

[1] The answer to this should be 1-2 lines of code. Keep it simple. If you struggle to do this, investigate different solutions to the previous questions that make this one easier.

[2] Do not hard code any numeric values.

ANS:

A string in C is an array of characters terminated by a NULL character. Strings will be null terminated by default.

- What happens when a program that is looking for a null terminator and does not find it.

ANS:

It will result in undefined behavior as you try to read past the end of the string.

- In this assignment, you are given a macro called `CLEAR_SCREEN`. What is it's data type? How and when do you use it?

ANS:

It is a string that is an ANSI escape code. It's data type is const char *. We will use it to clear the screen and we will simply use printf() to do so.

Testing

List what you will do to test your code. Make sure this is comprehensive. [3] Remember that you will not be getting a reference binary

ANS:

I will test the 256 char limit, different chars that are not allowed, multiple command line inputs, no command line inputs, functionality of the 4 functions, and exact outputs of specific cases. [4].

How to Use the Program

Audience: Write this section for the user of your program. You are answering the basic question, "How do I use this thing?". Don't copy the assignment exactly; explain this in your own words. This section will be longer for a more complicated program and shorter for a less complicated program. You should show how to compile and run your program. You should also describe any optional flags or inputs that your program uses, and what they do.

ANS:

Run "./hangman "secret"" on the command line to run the executable with a given secret.

Program Design

Audience: Write this section for someone who will maintain your program. In industry you maintain your own programs, and so your audience could be future you! List the main data structures and the main algorithms. You are answering the basic question, "How is this thing organized so that I can have a chance of fixing it?". This section will be longer for a more complicated program and shorter for a less complicated program.

[3] This question is a whole lot more vague than it has been the last few assignments. Continue to answer it with the same level of detail and thought.

[4] The output of your binary is not the only thing you should be testing!

ANS:

char *'s, arrays, strings, and integers are the data types used in the logic of the program. The 4 functions we define do most of the work, comparing down char * strings until the null terminator, checking if a character is a lowercase letter, and validating the initial secret and subsequent character guesses. While and for loops make up the majority of algorithm logic, as the Display is updated and guesses are executed.

Pseudocode

Give the reader a top down description of your code! How will you break it down? What features will your code have? How will you implement each function.

ANS:

```
check correct # of command arguments, validate the secret, initialize display components
while not at max bad guesses:
    print clear screen, the gallows, phrase, eliminated characters
    check for win or loss and end game accordingly, continue otherwise
    prompt for a new guess until a valid guess occurs
    update display or eliminated characters accordingly
```

Function Descriptions

For each function in your program, you will need to explain your thought process. This means doing the following

- The inputs of every function (even if it's not a parameter)
- The outputs of every function (even if it's not the return value)
- The purpose of each function, a brief description about a sentence long.
- For more complicated functions, include pseudocode that describes how the function works
- For more complicated functions, also include a description of your decision making process; why you chose to use any data structures or control flows that you did.

  ANS:

  bool string_contains_character(const char *s, char c) Takes s, a const char * as the string to check, and c as the character to find. The function will return true if the character is present and false otherwise. The characters are compared and s incremented until the character is found present or s is the null character.

  char read_letter(void) Takes no input and reads for a valid character input to return ignoring the newline char.

  bool validate_secret(const char *secret) takes a const char * secret as the string to check, and makes sure that it is 256 characters or less with only lowercase letters, spaces, hypens, and apostrophes. It will return true if the secret is valid and false otherwise.

  bool is_lowercase_letter(char c) takes a character c and returns true if it is a lowercase letter and falso otherwise.

Do not simply use your code to describe this. This section should be readable to a person with little to no code knowledge. **DO NOT JUST PUT THE FUNCTION SIGNATURES HERE. MORE EXPLANATION IS REQUIRED.**

References