Assignment 2 – Hangman Report Template

Nickolas Tran

CSE 13S – Winter 24

Purpose

For this task, the hangman game you create will undergo several modifications. Firstly, the secret word can consist of multiple words and may include characters such as apostrophes, spaces, and hyphens of your choosing with a maximum length of 256 characters. If the secret contains special characters, they will be revealed at the beginning of the game. During the game, the player is allowed to guess only one letter at a time. Until the amount of guesses is completed or the word has been solved.

Questions

- How many valid single character guesses are there? What are they?
  26 valid single character guesses. (A-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?
  Yes you do need to keep track of how many times something is guessed, so you can establish whether that person guessed it already. You don't need to track the order in which the user makes guesses unless you want to display the guessed letters.
- 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. [1]
  A good data type we can use is an "array".
- Based on your previous response, how can we check if a letter has already been guessed. [2]
  You can use an array to keep track of guessed letters. The array should be large enough to store the amount of guesses allowed and can store and show the player's guess.

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.
- Below is some python code. Finish the C code below so it has the same effect. [3]

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

C Code:

```
char x = 'q';
printf("%d\n", x);
```

[1] Your answer should not involve rearranging the old guesses when a new one is made.

[2] 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.

[3] Do not hard code any numeric values.

- 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.

```
#include <stdbool.h>
char is_uppercase_letter(char x){

}
```

- 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.
  A string is an array of characters. An example is if the string does not end with a null character.
- What does it mean for a string to be null terminated? Are strings null terminated by default?
  A string to be null terminated is a string stored as an array containing characters and terminated with a null character. Strings are null terminated by default.
- What happens when a program that is looking for a null terminator and does not find it.
  It might continue to read memory past the allocated space, which can lead to numerous issues and crashes.
- In this assignment, you are given a macro called `CLEAR_SCREEN`. What is it's data type? How and when do you use it?
  Macros are not data types. You use it to clear the terminal screen.

Testing

- Integrating testing: Making sure that the different functions are interacting together well.
- Unit testing: Making sure that the valid inputs are functional individually. Verifying that each function produces the expected output.
- Error testing: Making sure that the invalid inputs of the user is recorded and handles it. And checks when error messages are informative.
- UI Testing: Making sure that the UI is valid and responsive to the input of the user.

How to Use the Program

The user uses "make" in the command terminal to recompile the set of rules created by the program.

```
make
```

The user then uses "./hangman" to run the program.

```
./hangman <secret word>
```

Then the user started guessing the letters to the hangman.

Program Design

- Word Array: An array to store the letters of the target word.
- Guessed Array: An array to keep track of guessed letters.
- Hangman Drawing: ASCII art or a series of strings representing the hangman figure.

Pseudocode

```
int main():
    Declares arrays to store the secret phrase, eliminated letters , and guessed letters.
    Checks if the correct number of command-line arguments is provided; if not, the program exits
        with an error code.
    Copies the secret phrase from the command line argument, and validates it using the
        validate_secret function. If validation fails, the program exits with an error code.
```

```
validate_secret():
   if length of secret is greater than MAX_LENGTH:
       return false
    for each character in secret:
       if character is not a valid letter, space, dash, or apostrophe:
           return false
    return true
```

```
string_contains_character():
    for each character in input_string:
       if character is equal to target_character:
           return true
    return false
```

```
read_letter():
    letter = read a character from the user
    while letter is not a lowercase letter:
       print an error message
       letter = read a character from the user
    return letter
```

```
is_lowercase_letter():
   if c is between 'a' and 'z' (inclusive):
       return true
    return false
```

Function Descriptions

For function int main():

- Input: argc and argv
- Output: Return indicating 0 or non-zero for success or failure.
- Purpose: Initializes variables for the secret phrase, eliminated and guessed letters, and the hangman display. Validates the command-line argument and the secret phrase. Initiates the game loop, updating hangman display based on user input. Determines the win or lose condition and prints the result.

For function validate secret():

- Input: Secret phrase
- Output: Boolean indicating whether the secret phrase is valid
- Purpose: Ensures that the phrase doesn't exceed the maximum length.Checks if the secret phrase is non-empty and contains only valid characters.

For function string contains character():

- Input: Character or string.
- Output: Boolean indicating whether the character is present in the string.
- Purpose: Checks if a specific character is present in a given string.

For function read letter():

- Input: None
- Output: Character representing the letter read from the user.
- Purpose: Reads a single lowercase letter from the user and returns it.

For function is lowercase letter():

- Input: c
- Output: Boolean indicating whether c is a lowercase letter.
- Purpose: Checks if the given character is within the ASCII range of lowercase letters ('a' to 'z'). Returns true if it is, and false otherwise.