

# Assignment 2 – Hangman Report Template

Jayden Sangha

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.

## 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?

I think there should be 26 single character guesses, one for each letter of the English alphabet.

- 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, we should keep track of how many times a character is guessed, so players can not guess the same letter twice. However, I do not think we need to keep track of the order of the characters guessed, just which characters were guessed.

- 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. <sup>1</sup>

We can use an array to keep track of the player’s guesses.

- Based on your previous response, how can we check if a letter has already been guessed. <sup>2</sup>

When the player guesses we can iterate through the entire array to see if their input has already been guessed.

---

<sup>1</sup>Your answer should not involve rearranging the old guesses when a new one is made.

<sup>2</sup>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.

---

## 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.

The function `chr()` takes a number and converts it into a string character that correlates with the Unicode number entered. The function `ord()` is the opposite of the `chr()` function. You input a string character and the function outputs the Unicode number that corresponds with that letter.

- Below is some python code. Finish the C code below so it has the same effect. <sup>3</sup>

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

C Code:

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

- 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){
    if (x >= 'A' && <= 'Z')
        return true
    else
        return false
}
```

- 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.

In C a string is an array of characters that is terminated with a null character. C will not allow an empty string.

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

A string is null terminated when the last character is a null character like 0. Strings are automatically null-terminated unless the characters were initialized individually.

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

If the program does not find a null terminator it could cause several problems like messing up string lengths, unexpected outputs, or even issues with the memory.

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

The macro `CLEAR_SCREEN` is a string data type. We use this to clear the screen in the game by inserting the string into the macro.

## Testing

List what you will do to test your code. Make sure this is comprehensive. <sup>4</sup> Remember that you will not be getting a reference binary <sup>5</sup>.

We need to test that the player is inputting a valid character that is lowercase. We also need to test that the secret word isn't too long and make sure it contains valid characters. We also need to test that the game stops when the player runs out of guesses. We also should test that the player only enters one character. We need to test that the player's screen clears in between turns.

---

<sup>3</sup>Do not hard code any numeric values.

<sup>4</sup>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.

<sup>5</sup>The output of your binary is not the only thing you should be testing!

---

## 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.

First, you must compile the program by entering

```
make format hangman.c
```

then you must enter

```
./hangman
```

to start the program. Then it will prompt you to enter the secret word. After that, it will ask you to input your guesses. If you guess correctly that character of the word will appear, if you are incorrect a part of the man will appear. This will keep going until either the full word is guessed and the player wins or the whole man appears and the player loses.

To show “code font” text within a paragraph, you can use `\lstinline{}`, which will look like this: `text`.

For a code block, use `\begin{lstlisting}` and `\end{lstlisting}`, which will look like this:

Here is some code in `lstlisting`.

And if you want a box around the code text, then use `\begin{lstlisting}[frame=single]` and `\end{lstlisting}`

which will look like this:

```
Here is some framed code (lstlisting) text.
```

Want to make a footnote? Here’s how.<sup>6</sup>

Do you need to cite a reference? You do that by putting the reference in the file `bibtex.bib`, and then you cite your reference like this[1][2][3].

## 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.

## 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.

```
main function
```

```
create mistake counter
create max mistakes
run validate_secert
while mistake counter is less than max mistakes
run read_letter
run is_lowercase_letter
if false set max mistakes to 100

run string_contains_character
```

---

<sup>6</sup>This is my footnote.

---

```
if false mistake counter + 1
```

```
bool string_contains_character(const char* s, char c)
```

```
if char c not in char s
return true
else
return false
```

```
char read_letter(void)
```

```
create variable for input
prompts for character input
set input to variable
returns variable
```

```
bool is_lowercase_letter(char c)
```

```
if char c is greater than or equal to ascii for a and less than or equal to the ascii for z
return true
else
return false
```

```
bool validate_secret(const char* secret)
```

```
create variable for secret word
prompts for secret word
create variable length
set variable equal to strlen of secret
if length is greater than 256
print character limit error
return false
```

```
if not lowercase or invalid characters
print invalid character error
return false
```

## 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.

---

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.**

`bool string_contains_character(const char* s, char c)`

Its inputs are the secret word and the player's input. Its outputs are true if the letter is in the word and false if it is not. This function checks to see if the letter the player guessed is contained in the secret word.

`char read_letter(void)`

Its only input is the player's guess. Its output is the character the player inputted. This function prompts the player to input their guess.

`bool is_lowercase_letter(char c)`

Its input is the player's guess. It outputs true or false if the player's guess is upper or lower case. This function checks the player's input to make sure that it is lowercase.

`bool validate_secret(const char* secret)`

Its input is the secret word. It outputs true or false. This function checks that the secret word is less than 256 characters and that it is a valid character, if it is it will return true otherwise it will return false.

## References

- [1] Wikipedia contributors. C (programming language) — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/wiki/C\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/C_(programming_language)), 2023. [Online; accessed 20-April-2023].
- [2] Robert Mecklenburg. *Managing Projects with GNU Make, 3rd ed.* O'Reilly, Cambridge, Mass., 2005.
- [3] Walter R. Tschinkel. Just scoring points. *The Chronicle of Higher Education*, 53(32):B13, 2007.