

Hangman game

Cp theory project – spring 2024

**Group Members:**

1. Sana Ali | 01-134241-042
2. Ali Hassan | 01-134241-043

**Submitted To:** Sir Sohail Mohammad

**Submission Date:** 25/05/24

**Department:** Computer Science

**Class :** BS CS 1A

Contents

[Abstract 2](#_Toc166718537)

[Introduction 2](#_Toc166718538)

[Step by step explanation of the code 3](#_Toc166718539)

[Evaluation 9](#_Toc166718540)

[Conclusion 12](#_Toc166718541)

[Refrences 14](#_Toc166718542)

## **Abstract**

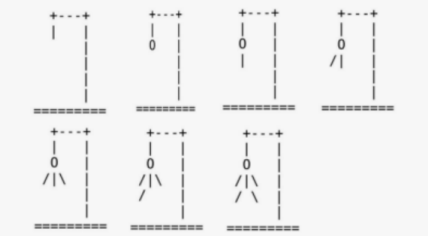
This report presents an evaluation of a Hangman game implementation using C++. The Hangman game is a classic word-guessing game where players attempt to guess a secret word letter by letter while avoiding losing all their lives, which are a total of 6. This report focuses on various aspects of the game's implementation, including game logic, user interaction, random word selection, outputs of the game, and code structure. Overall, the Hangman game implementation serves as a practical demonstration of fundamental programming concepts and provides a foundation for exploring further game development skills in C++.

## **Introduction**

The goal of this report is to briefly summarize the idea and basic rules of our Hangman Game that we made as our Computer Programming (Theory) Project.

In this game, the computer picks a random secret word from an available list. We will store a list of words using arrays while a random word will be chosen through the srand () function. The concept of the game is that the player then guesses a letter that they think may appear in the word. After guessing, the computer checks whether the letter appears in the word and either reveals all instances of it (if it exists) or takes away one of the player's lives. The computer then displays the progress that the player has made towards guessing the word. The game continues until either all of the letters have been guessed (and the player wins) or all of the player's lives are used up (and the player loses). In either case, the computer should announce the conclusion and the game should end.

As we know that the figure of hangman has 6 parts, hence the player will have 6 lives. We will be displaying each of these figures on our window once the letter is guessed and checked by the computer:

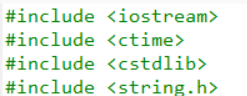


Thus, if the player guesses the correct word before all the lives are ov+er, the game terminates and a message is shown to the user that they have won as they guessed it right. Similarly, if all the lives are over and the player is still unable to guess our game will terminate and a message will be shown to the user that they have lost along with the correct word that they failed to guess.

## **Step by step explanation of the code**

Libraries Used:

Before we start with the explanation of our code, the libraries that we have used are the following:



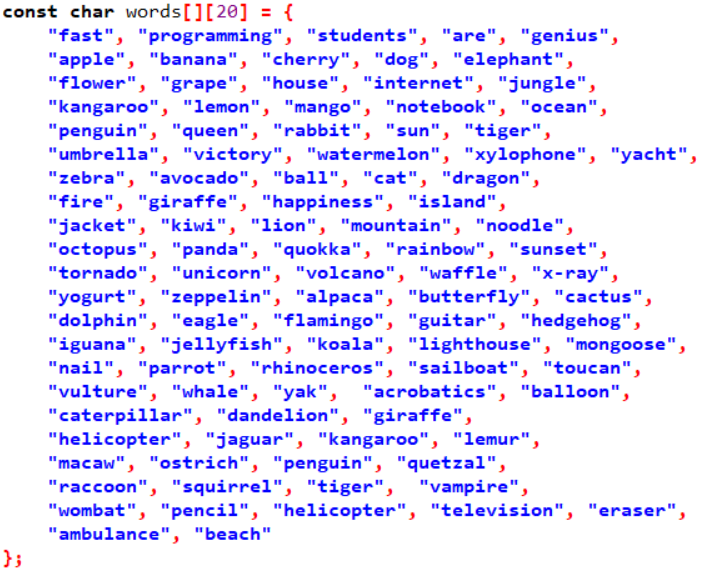
The <iostream> library is used for the input/output operations in c++, while <ctime> library is used to obtain the current system time. The <ctime> library is used along with <cstdlib> library that gives us srand() function to seed the random number generator. Similarly, the <string.h> library is used to get the inbuilt string functions such as strlen().

CONSTANT VARIABLES DECLARED BEFORE MAIN FUNCTION:

1. const int MAX\_LIVES = 6

Max lives are basically the 6 chances user gets to guess the word before the game is terminated.

1. A 2D array for the list of words we have declared:



1. And to find the numWords we use:

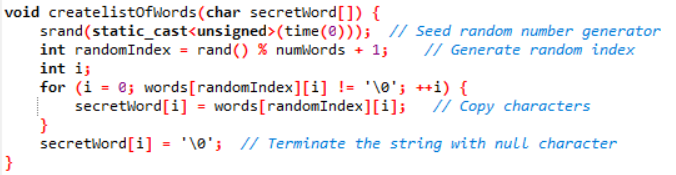


The expression sizeof(words) calculates the total size (in bytes) occupied by the entire words array. On the other hand, sizeof(words[0]) calculates the size (in bytes) of an individual element of the words array, which in this case corresponds to a single word string (a char array).

Thus, by dividing sizeof(words) by sizeof(words[0]), we obtain the number of elements (words) present in the words array.

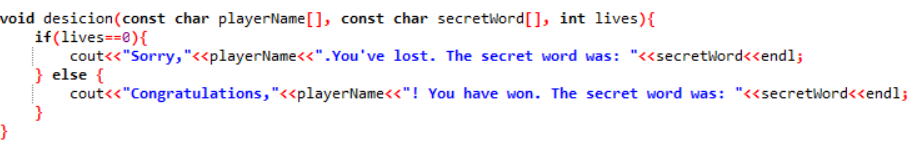
USER DEFINED FUNCTIONS:

1. Function to create list of words:



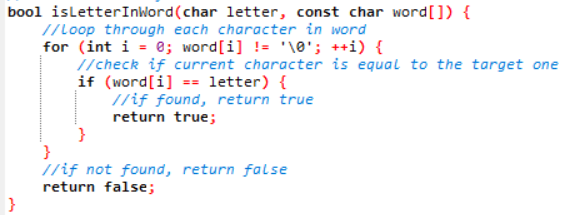
This function is in charge of transferring a randomly chosen word into the secretWord array from the predefined list (words). Based on the current time (time(0)), the random number generator is seeded using the srand() method. Subsequently, a random index (randomIndex) is generated within the words array's range, and characters from the chosen word are successively copied into secretWord until they reach the null terminator ('\0'), guaranteeing the string's proper termination.

1. Function to display the final outcome of the game:



When the game is over, this function is used to show the result. It takes three paramteres: lives, secretWord, and playerName. The secretWord is revealed along with a message indicating that the player has lost if all of their lives have been used up (lives == 0). If not, it shows the secretWord and congratulates the player on winning.

1. Function to check whether specified letter is present in the random word



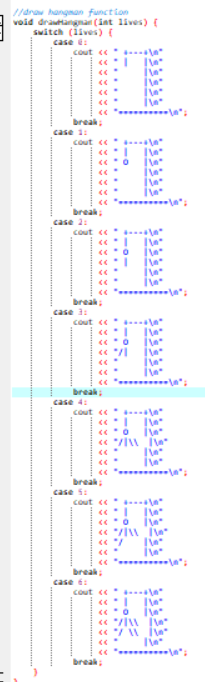
This function checks whether a specified letter is present in the word array. It iterates through each character of the word until it encounters the null terminator ('\0'). If the letter is found within the word, it returns true; otherwise, it returns false. This function is used during gameplay to validate the player's guesses.

1. Function to check if given character is alphabetic



This function is used for input validation to ensure that a given character (ch) is an alphabetic letter (either uppercase or lowercase). It checks whether the character falls within the ASCII range of alphabetic characters (A-Z or a-z) and returns true if it's alphabetic, false otherwise. This function is used basically for validating the player's name input to ensure it contains only alphabetic characters.

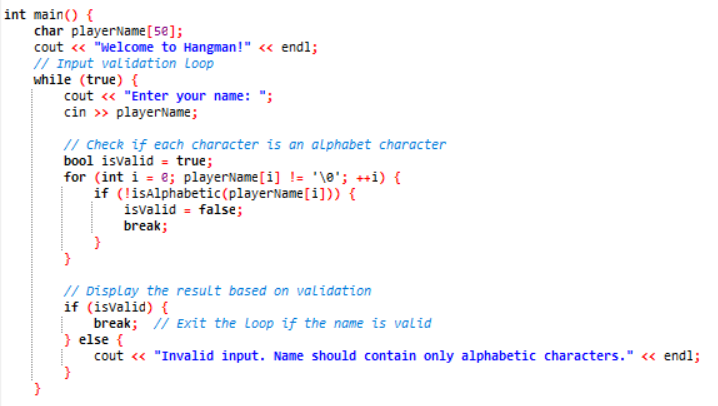
1. Function to draw Hangman based on the lives remaining



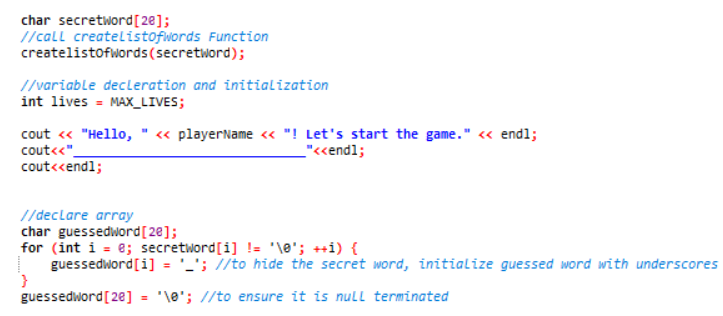
This function is responsible for drawing a hangman figure based on the number of lives remaining. It uses a switch-case statement to determine the appropriate hangman configuration for different values of lives. The drawHangman function visually represents the game progress, showing the hangman figure as the player makes incorrect guesses.

CODE INSIDE THE INT MAIN FUNCTION:

Now inside our int main function, we first of all take input of the players name as shown below in the snippet and do input validation by calling our “isAlphabetic” function to ensure the name entered is alphabetic (uppercase or lowercase). We do our input validation in a while(true) that is an infinite loop so that it continues to run until correct name is entered.



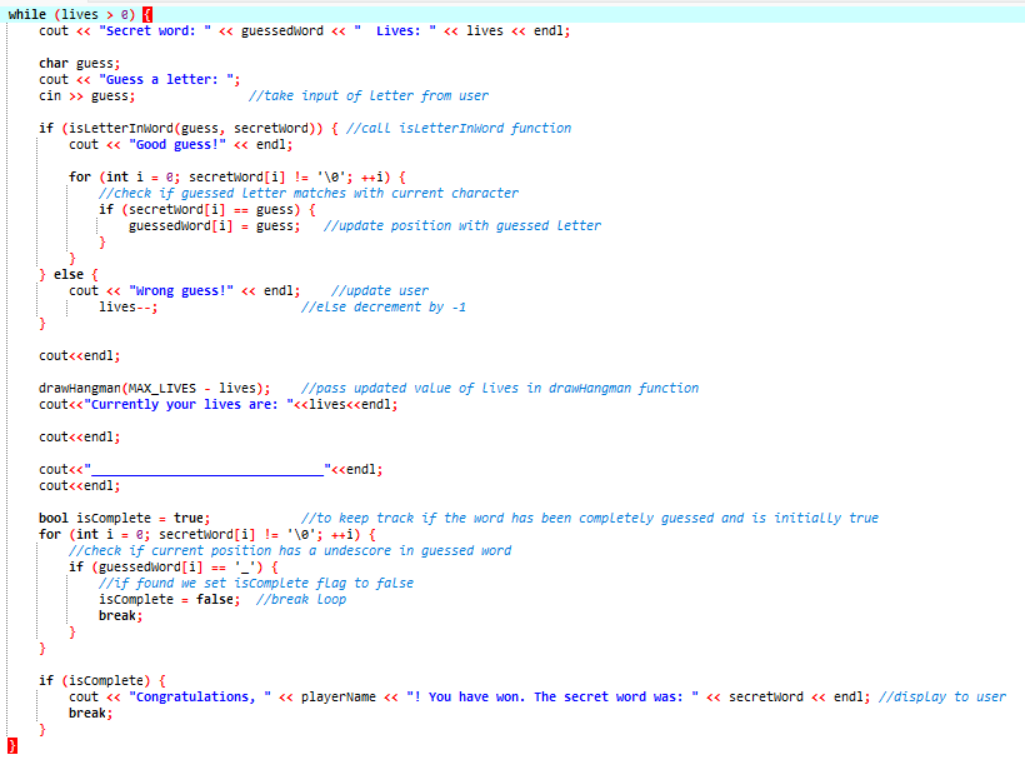
After the name has been given as input and checked we move on to our main code, as shown is snippet below:



We first initialize a character array called secretword and then call our “createlistOfWords” function in which the array is passed as parameter and it selects a random word from list in this function.

After this, another char array is initialized called guessedWord and a for loop is implemented to initialize our guessed word with underscores in order to hide our secret word. The loop continues until it reaches the null terminator.

Now the while loop where lives are greater then zero is main part of our int main as all basic functionality takes place there as shown below in snippet of code:



The while loop continues as long as the player's lives basically the chances to guess the word are greater than zero. During each iteration of the loop:

1.The current state of the guessed word (guessedWord) and the remaining lives are displayed to the player.

2.The player is prompted to enter a letter as a guess for the secret word.

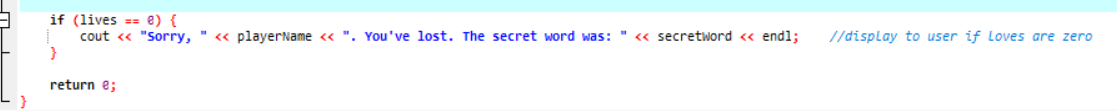
3.The program checks if the guessed letter is present in the secretWord using the “isLetterInWord” function. If the letter is correct, the program updates the guessedWord to reveal the correctly guessed letters.

4.If the guessed letter is incorrect, the program notifies the player and decrements the lives counter by one.

5.After processing the guess, the hangman drawing corresponding to the remaining lives is displayed using the drawHangman function.

6.The loop continues until either the player correctly guesses the entire word (isComplete is true) or runs out of lives (lives becomes zero).

After the loop has ended, we use an if statement to check if the lives are zero. If they are a message is displayed to user as shown in the snippet below:

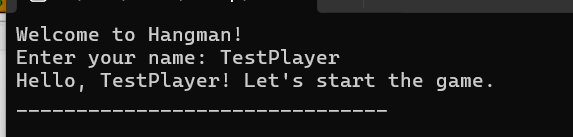


And this marks the end of our code for the Hangman Game made by us.

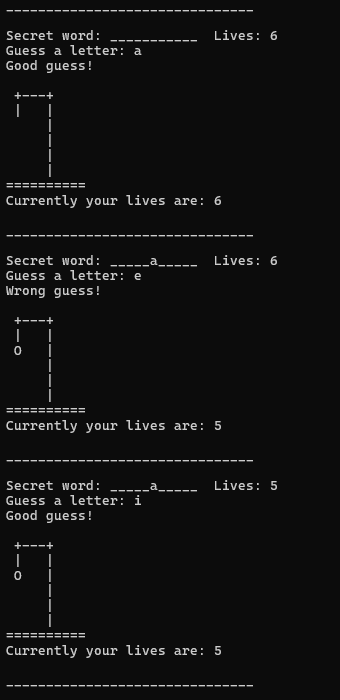
## **Evaluation**

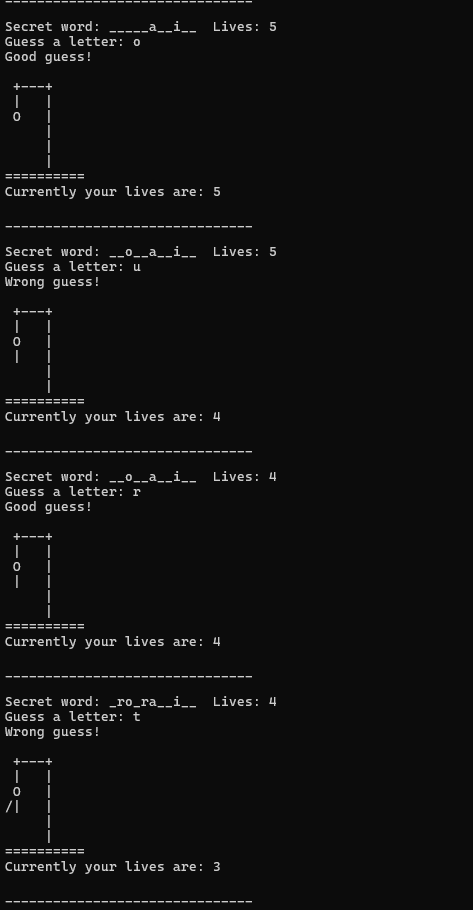
To evaluate our game, we can have a look at the output of our game.

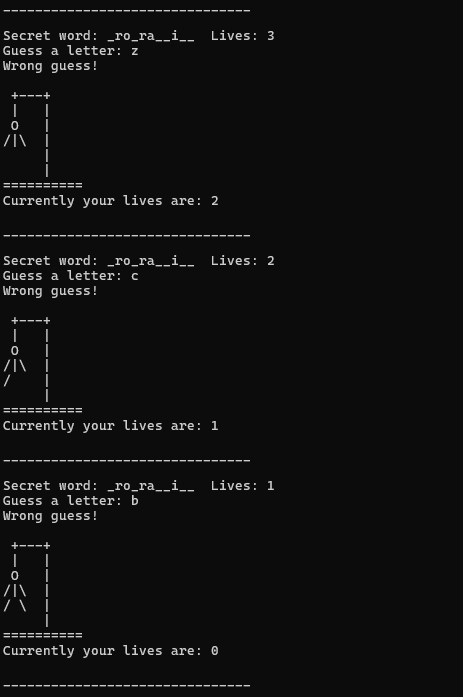
At start of the game, we first take input of name of player from user as shown below:



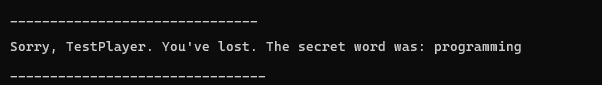
Now the rounds of guessing the words start, until the complete word is guesses or the user run out of lives.





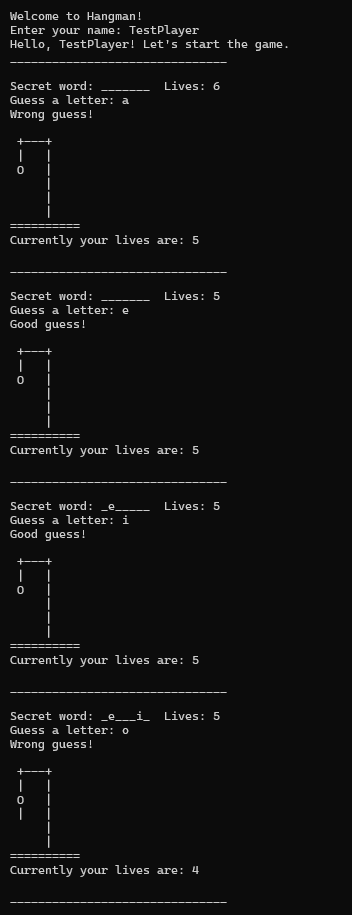


Once all the lives are over, the message is displayed for user on the window and the correct word that player failed to guess is displayed as well on the terminal window.



## **Conclusion**

To conclude our report, we have used C++ programming language, mainly focusing on the concepts of loops, arrays and function to make a successful Hangman Game. As shown by the outputs above our game is completely functional and correct. Furthermore, the Man structure made after each wrong guess makes the game even more interesting and fascinating. While the above output is shown of the case when the user is unable to guess properly, our code runs perfectly for the other case as well as shown below:





Thus, the output proves that our code runs perfectly in both cases that is if the player wins or runs out of chances and loses.

However, we can further improve the game by adding graphics and improving the user interface.

## **refrences and links**

Find Source code @GitHub: <https://github.com/Ali-Hassan-63>

:<https://github.com/SanaAli17>

Connect us @ Linked in: <https://www.linkedin.com/in/ali-hassan-4b92b12b5/>

We would like to acknowledge the invaluable assistance provided by MB academy, whose detailed video tutorial greatly contributed to the successful completion of this project. The video can be found at the following link: <https://www.youtube.com/watch?v=aSnvcU8prGk>