JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY



Odd Semester 2017 ( Semester III )

Data structures lab  
 (15B17CI371)  
 MINI PROJECT

hangman

SUBMITTED BY:  
 B-13 , Kirti Godani(168803003)  
 Shubham Aggarwal(16803006)  
 Swaraj Saxena (16803008)  
 Deepshi Sharma (16803021)

**Contents**

1. Problem Statement

2. Introduction

3. List of Data Structures used in the project

4. Detailed Design of the project

5. Implementation details and results

6. Conclusion

PROBLEM STATEMENT

Hangman is a paper and pencil guessing game for two or more players. One player thinks of a word and the other tries to guess it by suggesting the letters.The word to guess is represented by a row of dashes, giving the number of letters. If the guessing player suggests a letter which occurs in the word, the program writes it in all its correct positions. If the suggested letter does not occur in the word, the other player draws one element of the hangman diagram as a tally mark. The game is over when:  
  
**The guessing player completes the word, or guesses the whole word correctly.**

**HOW TO PLAY**: Our code will generate a word which has to be guessed by the player. So, at the output screen will exist marked out blanks (short lines) for each letter of a word. Then the player will guess a letter. If that letter is in the word(s) then the project will write the letter at everyplace it appears, and cross out that letter in the alphabet. If the letter isn't in the word then we cross out the lifelines (which are usually a finite no. of chances) from the list. The player will continue guessing the letters until he can either solve the word (or phrase) or he will end up losing all the lifelines and he will be declared a **LOSER.**

So, it is basically a TWO PLAYER game. But in my project a single player plays the game and the rules are strictly followed by the program.

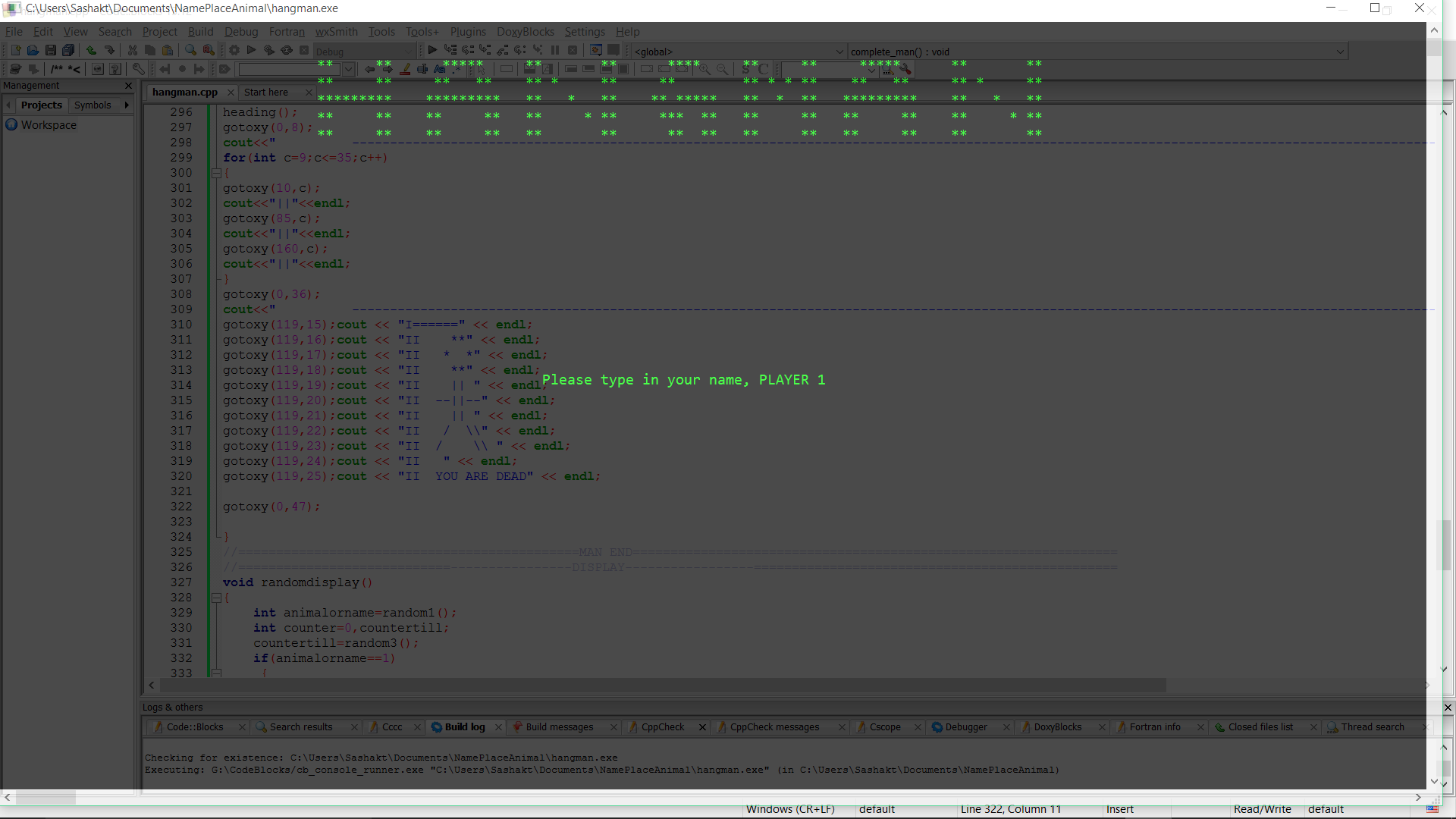
LIST OF DATA STRUCTURES USED

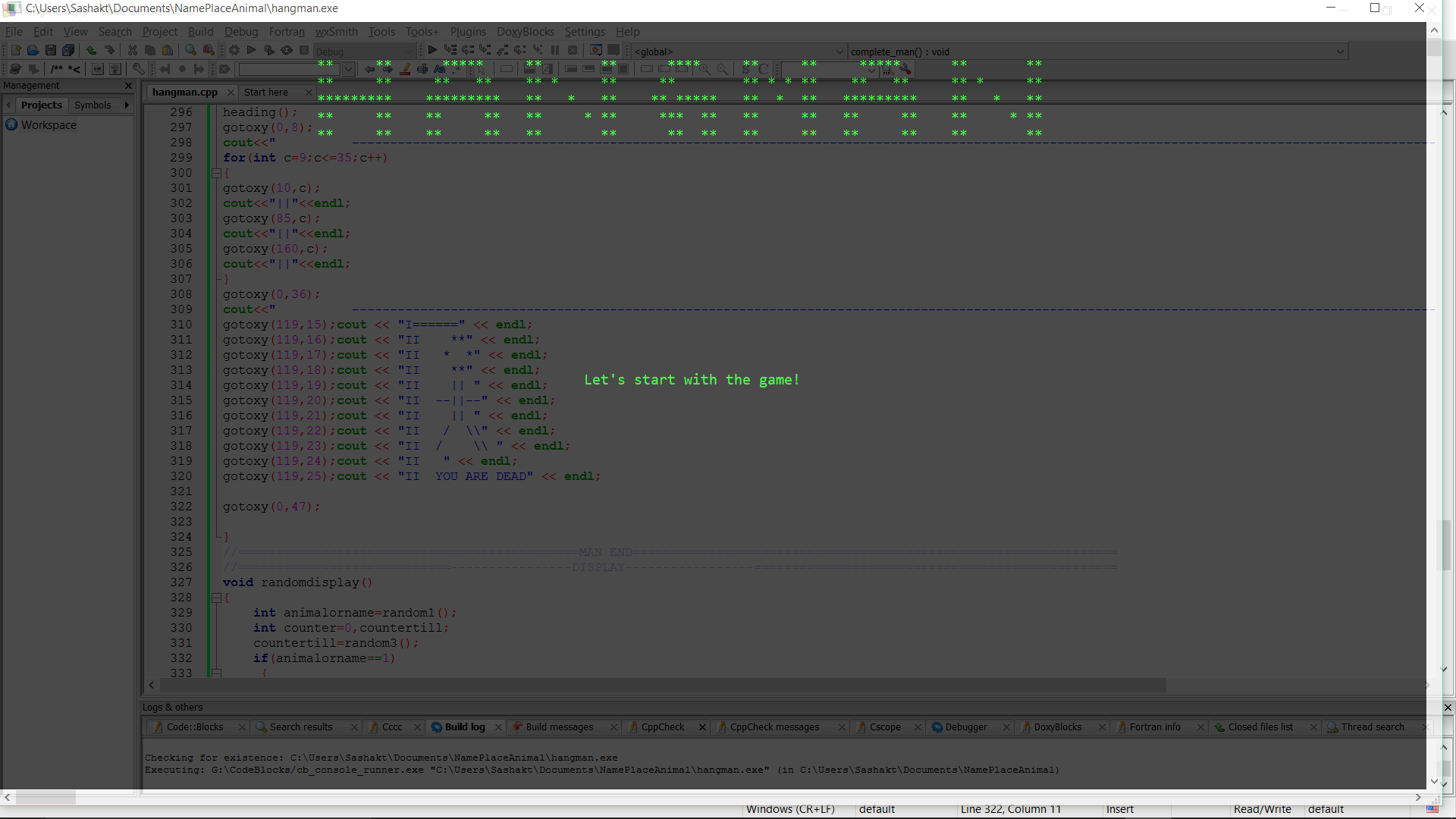
* LINKED LISTS
* DICTIONARY USING HASHING (CHAINING)

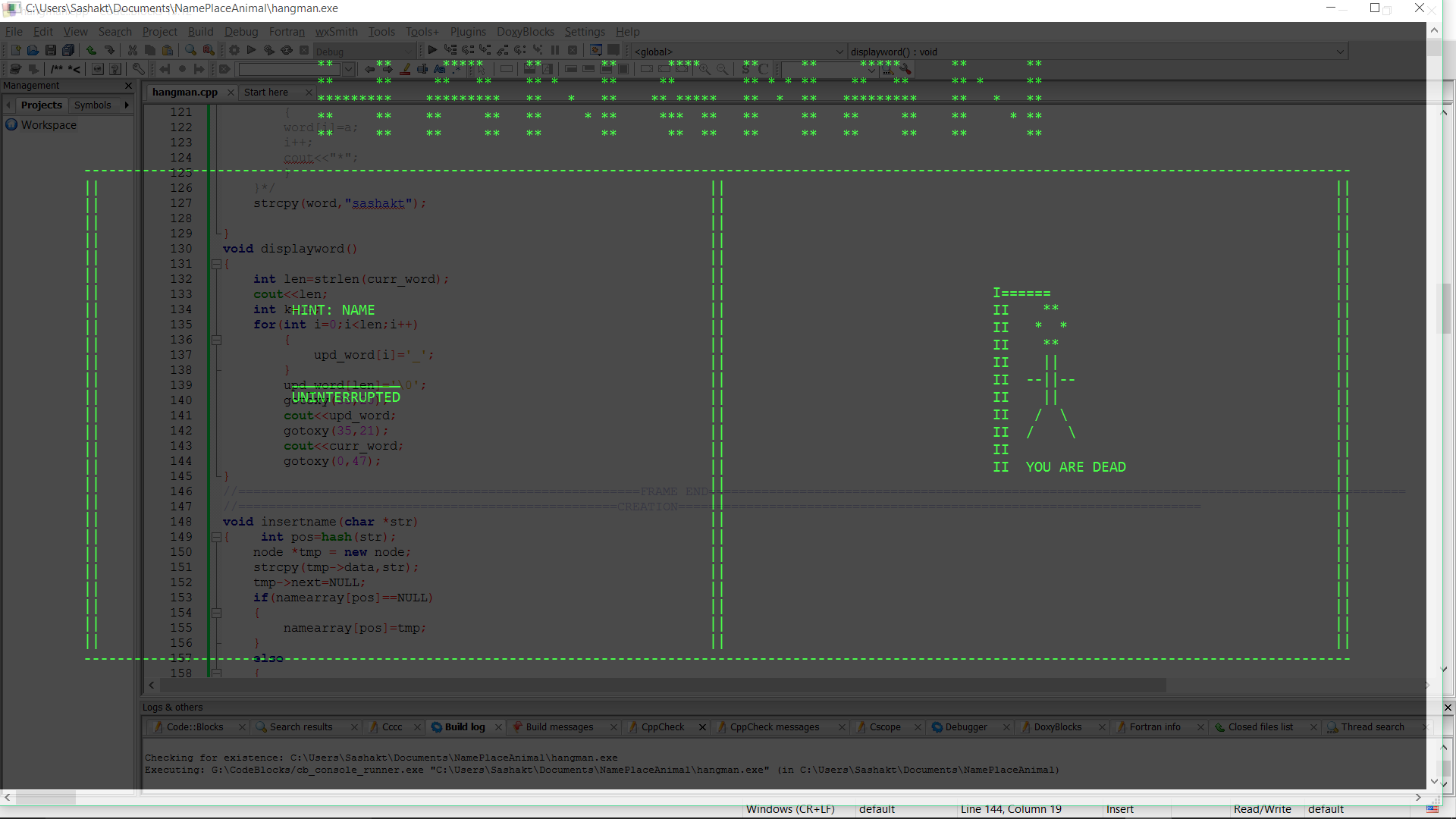
INTRODUCTION

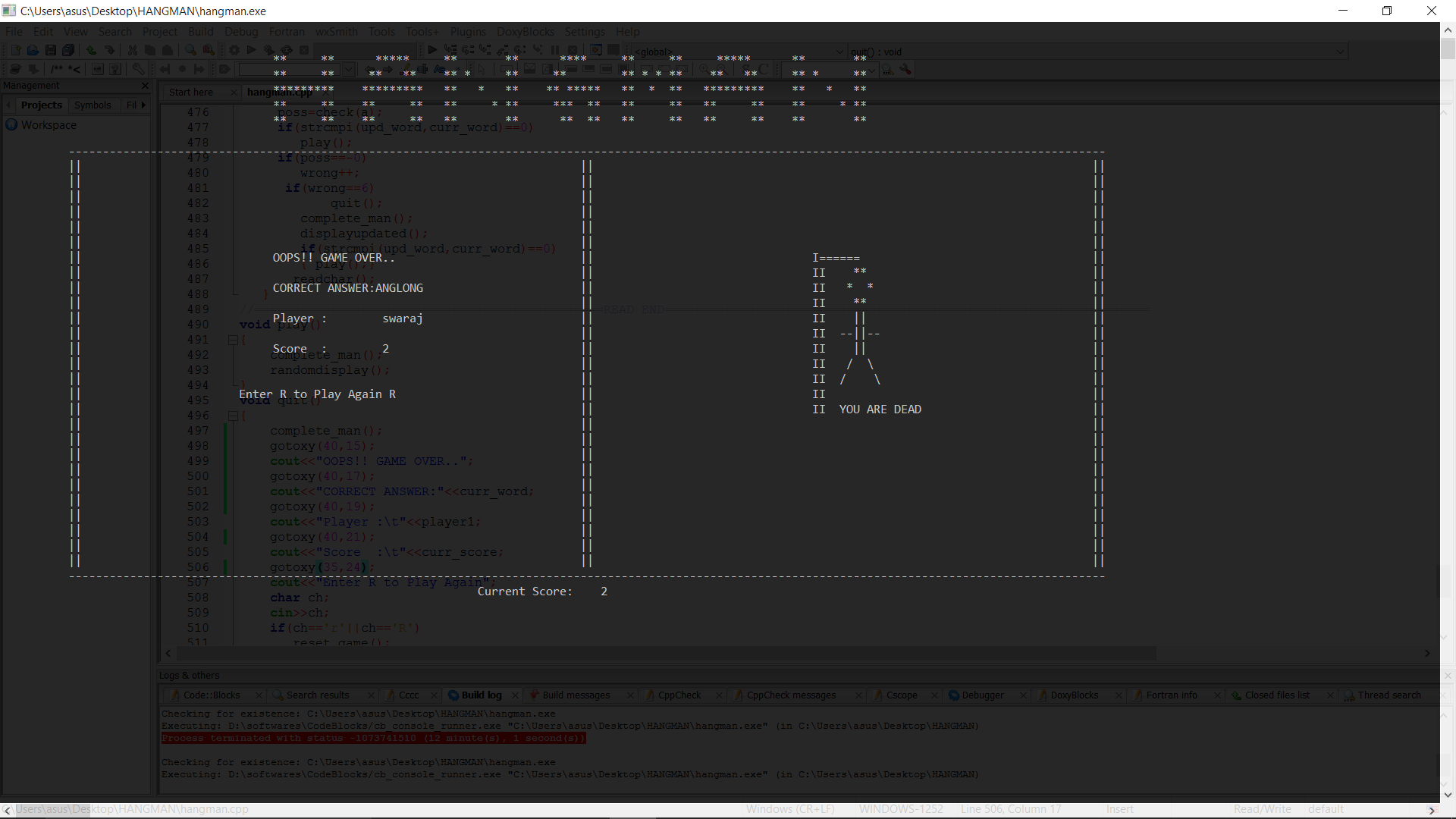
Hangman is a popular word game in which one player (the "chooser") chooses a secret word and another player (the "guesser") attempts to guess the word one letter at a time. If a guessed letter appears in the word, all instances of it are revealed. If not, the guesser loses a chance. If the guesser figures out the secret word before he or she runs out of chances, he or she wins. If not, the player who chose the word wins. Traditionally, chances are tracked using a stick figure drawing of a person being hanged from a gallows. The figure is drawn one body part at a time, and the guesser loses when the entire figure has been drawn. This game is also the basis for the TV game show Wheel of Fortune.

Design of the project









Implementation details and results

The gaming code will mainly contain the **class Hangman** which will provide the list of good letters as well as the no. of chances given to a user.Incorporation of some widgets will better the result screen , so that , the user can proceed in the game with no confusion. The overall architechture can be thought of having *four* main parts which consist of the following functionalities:-  
  
1) Formulating a word list (with or without a hint) and store them in a data structure with the list of all 26 alphabets of English Language.2) The actual method which does the logical reasoning , whether the letter exists or not , if yes , write it down at all the places else strike off a lifeline. 3)Final word to be displayed if guessed wrongly else, interactive message saying that "The Player is the winner"4)Finally,the coding , user interactive screen which will mainly prevail during the code output.

Conclusions

In conclusion, a program that can rapidly count the letter frequencies of a known word list can accurately guess a word very similar to a human, if not more so. Even if it can do so, the program has no insight nor can it have an epiphany as to what the word might be if the word is not contained in the list. From this we can draw that this program still has a long way to go in order to think like a human fully in a given task. To do so you just have to have more advanced algorithms, or humans with good metacognition.