**National University of Computer & Emerging Sciences**

**Karachi Campus**



**The Hangman Redefined**

**Project Report**

**Programming Fundamentals**

**Section: BCS1-J**

**Group Members**

**23k-2001 Muzammil Siddiqui**

**23k-3033 Uzair Haroon**

**23k-0800 Muhammad Mufeez**

**Introduction:**

We recreated the famous game we used to play in our childhood “The Hangman”, in C program with some additional rules to make it even more engaging than before.

We have worked to improve this nostalgic game with a touch of some new modes to make it far more captivating.

**Background:**

The famous game is available on the App Store, Play Store, and websites but is often limited to a similar category of words.

We proposed two management systems earlier but they were beyond our course outline. Also, we were advised to pursue a quiz or game program.

**Project Specification:**

Hangman refined with some add-ons.

### **The Hangman Classic**

## **Time trial**

## **Wordsmith Nemesis**

**Problem Analysis:**

The hangman we used to play often gets boring when the word we guessed before constantly pops up. This not only made it repetitive but also failed to capture the attention and interest of the user for a long time. The game usually does not contain the feature to choose the

topic they want the words from. They must guess random words without any context.

**Solution Design:**

We worked on vast arrays depending upon selected categories, which contain ample amounts of vocabulary to keep it as engaging as possible. There will be sub-categories for each genre which further contain difficulty levels.

We added three modes using switch statements:

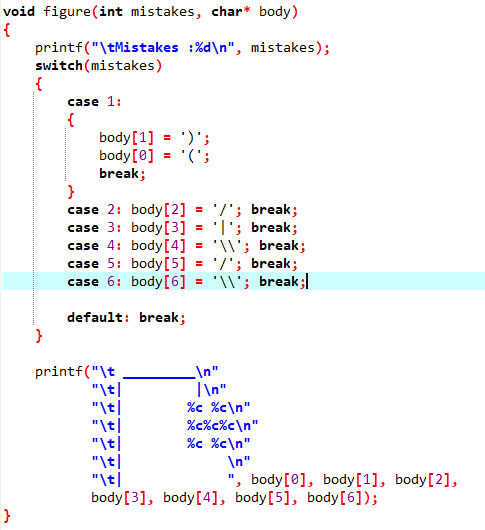
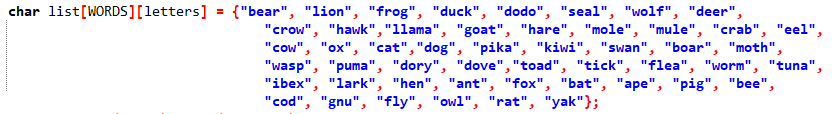
* **The Hangman Classic**
  + A usual word guess game we use to play. Complete all dashes by guessing letters before the stick figure gets hanged!
* **Time Trial**
  + Challenge yourself to get as many words as possible in a limited time.
* **Wordsmith Nemesis**
  + Complete the word and push your car to the finish line competing with a bot.

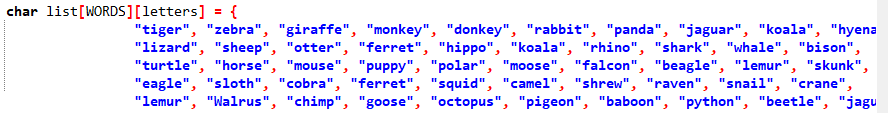
**Extra Features:**

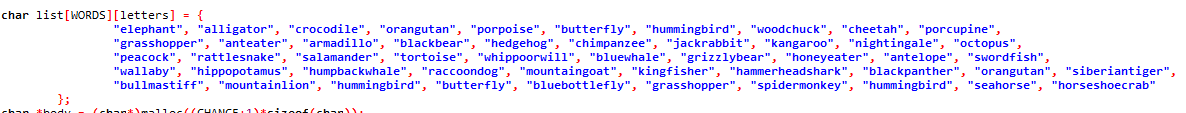
* Difficulty Levels- Challenge your limits!
* Time limit. Beat the clock.
* Categories to choose from, for example: Animals, Countries, food, random.

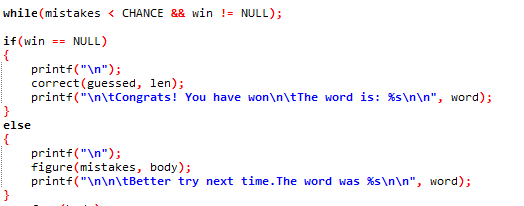
**Implementation & Testing:**

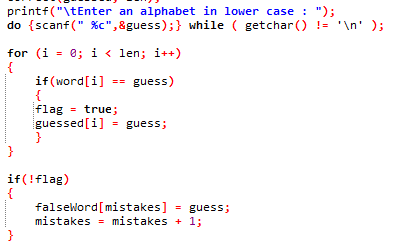
* The welcome screen displays rules and modes allowing the user to interact and proceed accordingly. 3 modes with 4 vocabulary categories with 3 difficulty levels.

* Hanged body printing mechanism
* 50 WORDS each with below 4-5 letters as normal 5 above and below 7-8 as medium and 9 above as hard





* Declaring result mechanism with the help of while loop and conditional ifs to check for condition till mistakes are less than chances available and the word is incomplete. When the word is completed before chances run out the user is declared the winner and otherwise in other cases.
* Storing the incorrectly guessed letters in a falseword[] array and incrementing mistakes using flag variable. The len used in for loop is to ensure the loop till the number of letters in a word



* Mechanism for time control. 60 seconds to guess in time trial mode

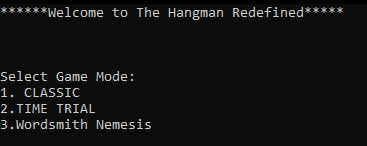
**Project workload distribution:**

At first, we made the structure of the project(nested switch of mode, categories, difficulty). And we divided each mode to each group member. But there was no such progress. Each one of us started working on mode 1 right after mid-2.

The difficulty Uzair faced was finding appropriate categories and 50 words for each. The categories were made according to the number of letters in each word. Uzair has to find common, yet hard-to-guess words.

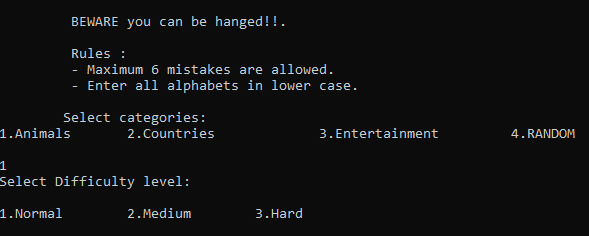
Meanwhile, muzammil started working on mode “wordsmith nemesis” from mid of November. Right after categories, Uzair and Mufeez started working on time trial mode from 19 November.

The most difficult part was the starting point, the main structure, and the time management as we started working just after mid-2. The testing was difficult as each time we had to play the game from the start repeatedly to make sure the code was directed in the way intended by us.

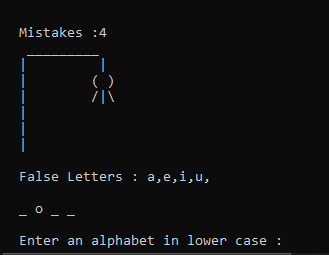
**Result (Output Screenshots):**

MENU:

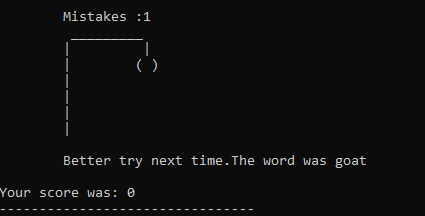
### Rules, Categories, and difficulty:

****

### Classic Mode:



### Time Trial:

****

### 

### Wordsmith Nemesis:

**Conclusion:**

This was our first-ever project where we implemented the things we learned in depth. The project required lots of effort and took over a month to complete. But still, there is always room for improvement. In the future, we can modify the program by adding more features such as hints for each word, and a leaderboard system. We can even add a graphics library for the mode ”Wordsmith Nemesis”, where we will be able to implement the car race in better graphics rather than ASCII art on the black console to make it even more engaging.