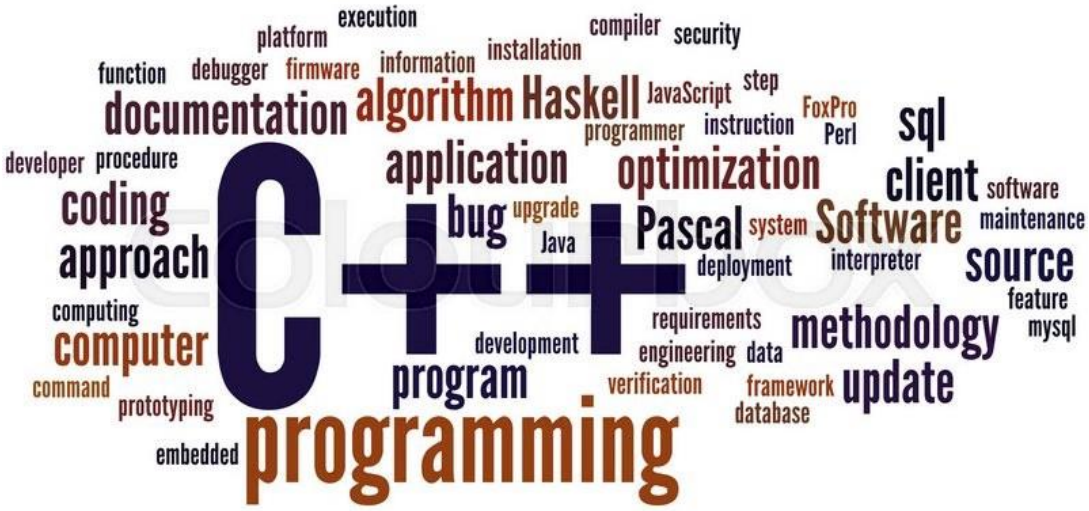
HANGMAN



DONE BY:- CHIRAG.K.PARIKH

C++ HEADER FILES USED

1. #INCLUDE<IOSTREAM.H>
2. #INCLUDE<CONIO.H>
3. #INCLUDE<STRING.H>
4. #INCLUDE<CTYPE.H>
5. #INCLUDE<PROCESS.H>
6. #INCLUDE<STDLIB.H>
7. #INCLUDE<IOMANIP.H>

C++ USER DEFINED FUNCTIONS PROTOTYPES

1. void word();
   - Contains all the words of different categories(Animals, Sportspersons and TV Shows). This function also selects a random word from the category chosen and passes the word to main_game().
2. void menu();
   - Displays the menu for the player and calls the appropriate function based on the player's choice.
3. void midscreen();
   - A function which allows the output to be shown in the middle of the screen.
4. void main_game(char a[50]);
   - It contains the main logic of the game. This takes a character from the player and checks if it is present in the word(the word is a string which is to be guessed). If it is present, then it converts that particular letter in the word to uppercase. If it is not present, it shows a TRY AGAIN error. A wrong guess is allowed for six or seven times, as is the case in HANGMAN. While printing it prints

only uppercase letters and prints a '_' when it encounters a lowercase letter.

A small example: -

Assume the word "shiva" is stored in the array.

If the user enters 'i', then the word gets converted to "shlva".

As mentioned above, while printing, it checks and prints only uppercase characters and '_' in case of lowercase characters.

So, while printing, it gives the output: - _ _ I _ _

A wrong guess of an alphabet will increment a variable and the program will stop when it reaches six or seven.

5. void how_to_play();
   - This function contains necessary information which will help the player in understanding how to play the game.
6. void settings();
   - This function contains settings which can be accessed by the player.
7. void credits();
   - This function is executed first when the program is run. It shows information like name of the game and the creator.
8. void category();
   - This function shows a menu of the different categories available from which the player can select a word for him to be guessed.

STRUCTURES CREATED

1. struct score
   {
   int correct,wrong;
   }s1;
   - Created for counting the overall score of the player. Correct contains number of words guessed successfully and wrong contains number of words that the player couldn't guess.

Note: For more details, you can refer the readme of the different versions.

THE END