**Word Find Game**

**Main idea**

The main idea of our project was to create a user-friendly game which mimics the game “Word Find”. In this game, the player will be provided with a combination of four letters, and then the player will have to make three and four letter meaningful words to continue.

**Main features**

The main features of the project which implement the original idea are as follows:

* Generating four random letters.
* Extracting three letters from those four random letters.
* Checking all the possible combinations of those randomly generated letters in the file, where we stored four and three letters word dictionary, to see if the meaningful words can be formed from those letters or not.
* Storing the meaningful words in an array.
* Verifying that the user enters the correct word.

**Screenshots**



*Explanation*:

The above mentioned snippet shows all the libraries, prototypes of functions and global variables used in our program.



*Explanation:*

In main function 4 random letters are generated and stored in string. The string is then passed to the function “PermuteFourLetter” and “ThreeLetterCombinations” which performs permutation on string.



*Explanation:*

The 4 letter word is taken as an input and is checked in the array, where all possible words are stored that can be made from the randomly generated string. If the entered word is correct, player is asked to enter the next 4 letter word. And if it is not correct then the player is asked to retry. It is also checked here that the user do not enter the correct word again.



*Explanation:*

As the player succeeds in entering correct two 4 letter words, player is asked to enter 3 letter word that can be made from the jumbled words. Then it performs the same checking which was done for 4 letter word.



*Explanation:*

Checks if the player wants to play the game again or not.

 

*Explanation:*

The above snippet is the definition of “PermuteFourLetter” function. This function is doing permutation on the 4 letter word string and it is also calling swap function to permute. Permutation is done using recursion. After each permutation, the text file, The Four Letter Word Dictionary is opened and permuted string is compared with the words stored in the text file line by line. If the comparison is true then the count is incremented and the string is stored in a 2D-array, string\_4. And if the comparison is false then the next permuted string is checked. In addition, if the string has two same letters for example in “PEEP” letter ‘E’ occurs two time then during permutation the same string will be generated twice and the comparison will come out to be true due to which two same letters will be stored in an array. To avoid storing the same word twice we first checked the 2D-array, if the word is already present there then same word will not be stored in an array.



*Explanation:*

The above snippet is the definition of “swap” function. This function simply performs swapping of characters using pointer. It is called in PermuteFourLetter and PermuteThreerLetter function.



*Explanation:*

The above snippet is the definition of “ThreeLetterCombinations” function. It performs the manual combination. The three letter string is generated from the randomly generated four letter string, and is stored in different arrays. The PermuteThreeLetter function is called in this function and the array is passed as an argument.

 

*Explanation:*

The above snippet is the definition of “PermuteThreeLetter” function. It performs permutation on three letter string.