**Question**

A class Rearrange has been defined to modify a word by bringing all the vowels in the word at the beginning followed by the consonants.

Example:  ORIGINAL becomes OIIARGNL

Some of the members of the class are given below:

Class name: Rearrange

Data Member/instance variable:

word: to store a word

newword: to store the rearranged word

Member functions/methods:

Rearrange(): default constructor

void readword(): to accept the word in UPPER case vow

freq\_vow\_con(): finds the frequency of vowels and consonants in the word and displays them with an appropriate message

  void arrange(): rearranges the word by bringing the vowels at the beginning followed by consonants

void display(): displays the original word along with the rearranged word

Specify the class Rearrange, giving the details of the constructor(), void readword ,void freq \_vow\_con(), void arrange() and void display(). Define the main() function to create an object and call the functions accordingly to enable the task.

**Algorithm**

1. Start

2. Define a class `Rearrange` with the following instance variables:

- `word`: a string to store the original word.

- `newword`: a string to store the rearranged word.

3. Define a default constructor for the class `Rearrange`:

- Initialize `word` to an empty string.

- Initialize `newword` to an empty string.

4. Define a method `readword()` for the class `Rearrange`:

- Create a `Scanner` object to read input from the user.

- Prompt the user to enter a word in UPPER case.

- Read the word from the user and store it in `word`.

5. Define a method `freq\_vow\_con()` for the class `Rearrange`:

- Initialize `vowelCount` to 0.

- Initialize `consonantCount` to 0.

- Define a string `vowels` containing the characters 'A', 'E', 'I', 'O', 'U'.

- Loop through each character in `word`:

- If the character is found in `vowels`, increment `vowelCount`.

- Otherwise, increment `consonantCount`.

- Print the frequency of vowels.

- Print the frequency of consonants.

6. Define a method `arrange()` for the class `Rearrange`:

- Define a string `vowels` containing the characters 'A', 'E', 'I', 'O', 'U'.

- Create a `StringBuilder` object `vowelPart` to store the vowel part of the word.

- Create a `StringBuilder` object `consonantPart` to store the consonant part of the word.

- Loop through each character in `word`:

- If the character is found in `vowels`, append it to `vowelPart`.

- Otherwise, append it to `consonantPart`.

- Concatenate `vowelPart` and `consonantPart` to form `newword`.

7. Define a method `display()` for the class `Rearrange`:

- Print the original word.

- Print the rearranged word.

8. Define a `main` method for the class `Rearrange`:

- Create an object of the class `Rearrange`.

- Call the `readword()` method on the object to accept a word from the user.

- Call the `freq\_vow\_con()` method on the object to count and display the frequency of vowels and consonants.

- Call the `arrange()` method on the object to rearrange the word.

- Call the `display()` method on the object to display the original and rearranged words.

9. End