**Question**

A class Encode has been defined to replace only the vowels in a word by next corresponding vowel from a new word.

Input – institution

Onstotation

Some of the member of the class are given below:

Class name: Encode

Data members

word: to store the word

length: integer to store the length of the word

new\_word: to store the changed word

Member method

Encode()- default constructor to initialize data members with legal initial valued.

Void acceptWord()- to accept a word.

Void nextVowel()- to replace only the vowels from the word stored in word by next corresponding vowel and to assign to new \_Word with remaining alphabets changed.

Void display()- to display the original word along with encrypted word.

Specify the class Encode giving details of the contructor(),void acceptWord(), void nextVowels() and void display().Define a main method to create an object and call the function and accordingly to enable the task.

**Algorithm**

1. Start

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

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

- `len`: an integer to store the length of the word.

- `newWrd`: a string to store the changed word.

3. Define a constructor for the class `Encode`:

- Initialize `wrd` as an empty string.

- Initialize `len` to 0.

- Initialize `newWrd` as an empty string.

4. Define a method `acceptWord()` for the class `Encode`:

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

- Prompt the user to enter a word.

- Read the word from the user and convert it to uppercase.

- Set `wrd` to the entered word.

- Set `len` to the length of the word.

5. Define a method `freqVowCon()` for the class `Encode`:

- Initialize a counter `count` to 0.

- Loop through each character in the word:

- If the character is a vowel ('A', 'E', 'I', 'O', 'U'), increment the counter.

- Print the frequency of vowels.

6. Define a method `nextVowel()` for the class `Encode`:

- Loop through each character in the word:

- If the character is 'A', append 'E' to `newWrd`.

- If the character is 'E', append 'I' to `newWrd`.

- If the character is 'I', append 'O' to `newWrd`.

- If the character is 'O', append 'U' to `newWrd`.

- If the character is 'U', append 'A' to `newWrd`.

- If the character is not a vowel, append the character itself to `newWrd`.

7. Define a method `disp()` for the class `Encode`:

- Print the original word.

- Print the encrypted word.

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

- Create an object of the class `Encode`.

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

- Call the `freqVowCon()` method on the object to count and display the frequency of vowels.

- Call the `nextVowel()` method on the object to encrypt the word.

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

9. End

**Variable Description Table**

|  |  |  |
| --- | --- | --- |
| **Variable name** | **Type** | **Description** |
| wrd | String | This variable stores the word entered by the user. It is used to hold the original word that needs to be processed |
| len | int | This variable stores the length of the word entered by the user. It is used to iterate through the characters of the word. |
| newWrd | String | This variable stores the encrypted version of the original word. It is constructed by replacing vowels with their subsequent vowels in the sequence. |
| count | int | This variable is used to count the number of vowels in the original word. It is incremented each time a vowel is encountered during the iteration. |
| ch | char | This variable is used to store the character at the current position during iteration through the word. It is used to check if the character is a vowel and to perform the vowel replacement. |