Started on	Tuesday, 10 October 2023, 1:52 PM
State	Finished
Completed on	Thursday, 12 October 2023, 1:09 PM
Time taken	1 day 23 hours
Marks	3.00/3.00
Grade	**15.00** out of 15.00 (**100**%)
Name	BALAJI S CSD

Question **1**

Correct

Mark 1.00 out of 1.00

As a logic building learner you are given the task to extract the string which has vowel as the first and last characters from the given array of Strings.

Step1: Scan through the array of Strings, extract the Strings with first and last characters as vowels; these strings should be concatenated.

Step2: Convert the concatenated string to lowercase and return it.

If none of the strings in the array has first and last character as vowel, then return no matches found

input1: an integer representing the number of elements in the array.

input2: String array.

Example 1:

input1: 3

input2: {“oreo”, “sirish”, “apple”}

output: oreoapple

Example 2:

input1: 2

input2: {“Mango”, “banana”}

output: no matches found

Explanation:

None of the strings has first and last character as vowel.

Hence the output is no matches found.

Example 3:

input1: 3

input2: {“Ate”, “Ace”, “Girl”}

output: ateace

For example:

Input	Result
3 oreo sirish apple	oreoapple
2 Mango banana	no matches found
3 Ate Ace Girl	ateace

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class Vowel
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        String arr[]=new String[n];
        for(int i=0;i<n;i++)
        arr[i]=sc.next();
        for(int i=0;i<n;i++)
        arr[i]=arr[i].toLowerCase();
        int c=0;
        for(int i=0;i<n;i++)
        {
```

```
            if(checkvowel(arr[i])==1)
            {
                System.out.print(arr[i]);
                ++c;
            }
        }
            if(c==0)
            System.out.println("no matches found");
        }
        public static int checkvowel(String s)
        {
            char x=s.charAt(0);
            char y=s.charAt(s.length()-1);
            if(x=='a'||x=='e'||x=='i'||x=='o'||x=='u')
            return 1;
            else
            return 0;
        }
    }
```

	Input	Expected	Got	
✔	3 oreo sirish apple	oreoapple	oreoapple	✔
✔	2 Mango banana	no matches found	no matches found	✔
✔	3 Ate Ace Girl	ateace	ateace	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **2**

Correct

Mark 1.00 out of 1.00

You are provided with a string which has a sequence of 1's and 0's.

This sequence is the encoded version of a English word. You are supposed write a program to decode the provided string and find the original word.

Each alphabet is represented by a sequence of 1s.

This is as mentioned below:

A : 1

B : 11

C : 111

D : 1111

E : 11111

F : 111111

G : 1111111

and so on upto Z having 26 1's (11111111111111111111111111).

The sequence of 1's in the encoded form are separated by a single 0 which helps to distinguish between 2 letters.

Example 1:

input1: 101101110

The decoded string (original word) will be: ABC

Example 2:

input1: 11111111011111011111111111101111111111110111111111111111110

The decoded string will be: HELLO

Note: The decoded string must always be in UPPER case.

For example:

Input	Result
101101110	ABC
11111111011111011111111111101111111111110111111111111111110	HELLO

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class Word
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        String s=sc.nextLine();
        char m[]=s.toCharArray();
        int c=0;
        char ch[]={'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U
        for(int i=0;i<s.length();i++)
        {
            if(m[i]!='0')
            {
                c++;
            }
            else
            {
                System.out.print(ch[c-1]);
                c=0;
            }
        }
    }
}
```

	Input	Expected	Got	
✔	101101110	ABC	ABC	✔
✔	11111111011111011111111111011111111111101111111111111110	HELLO	HELLO	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Question **3**

Correct

Mark 1.00 out of 1.00

Given a **non-empty** string s and an abbreviation abbr, return whether the string matches with the given abbreviation.

A string such as "word" contains only the following valid abbreviations:

["word", "1ord", "w1rd", "wo1d", "wor1", "2rd", "w2d", "wo2", "1o1d", "1or1", "w1r1", "1o2", "2r1", "3d", "w3", "4"]

Notice that only the above abbreviations are valid abbreviations of the string "word". Any other string is not a valid abbreviation of "word".

Note:

Assume s contains only lowercase letters and abbr contains only lowercase letters and digits.

Example 1:

Input

internationalization
i12iz4n

Output

true

Explanation

Given **s** = "internationalization", **abbr** = "i12iz4n":

Return true.

Example 2:

Input

apple
a2e

Output

false

Explanation

Given **s** = "apple", **abbr** = "a2e":

Return false.

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;
public class Solution
{
    public static boolean validWordAbbreviation(String word, String abbr)
    {
        int i = 0, j = 0;
        while (i < word.length() && j < abbr.length())
        {
            if (abbr.charAt(j) >= '0' && abbr.charAt(j) <= '9')
            {
                int num = 0;
                while (j < abbr.length() && abbr.charAt(j) >= '0' && abbr.charAt(j) <= '9')
                {
                    num = num * 10 + abbr.charAt(j) - '0';
                    j++;
                }
                i += num;
            } else
            {
                if (word.charAt(i) != abbr.charAt(j))
                {
                    return false;
                }
                i++;
                j++;
            }
        }
        return i == word.length() && j == abbr.length();
    }

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        String word = sc.nextLine();
        String abbr = sc.nextLine();

        boolean result = validWordAbbreviation(word, abbr);
        System.out.println(result);
    }
}
```

	Input	Expected	Got	
✔	internationalization i12iz4n	true	true	✔
✔	apple a2e	false	false	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

◄ WEEK_07_MCQ

Jump to...

WEEK_08_MCQ ►