Question - 1
Fantabulous Arrays

A *fantastic* array is one in which each new element in the array is greater than the sum of all the previous elements. For example, { 2, 3, 6, 13} is a *fantastic* array while { 2, 3, 5, 11} is not a *fantastic* array.

A *fabulous* array is one in which each new element in the array is greater than the product of all the previous elements. For example, { 2, 3, 9, 88} is a *fabulous* array while { 2, 3, 6, 54} is not a *fabulous* array.

A array is said to be *fantabulous* if it is both fantastic and fabulous.

Write a program to check if a given array of integers is fantastic, fabulous or fantabulous. Your program should output one of the below according to the type of the array:

- Fantastic Array
- Fabulous Array
- Fantabulous Array
- Array (if the array is none of the above)

Question - 2
Maximum Span

In an array, a span of an element is the number of elements between the first and the last occurrence of the element in the array, including the first and last occurrences of the element.

Write a program to find the largest span in a given integer array.

Question - 3
Identify duplicates

In the list of given inputs, identify the duplicates.
example input (First line should mention how many inputs):
5
inautix
technologies
private
limited
private

Output:
PRIVATE

Example Input-2:
4
LIMITED
PRIVATE
limited
private

PRIVATE

Question - 4
Compression

Given a string, *str*, of length *n* composed of English letters (*a-z,A-Z*), a *compression operation* on *S* is described as follows:

1. Find a substring having some repeated character, *c*. The substring *must* contain every consecutive repetition of *c* for that section of *str* (i.e.: the left and right boundaries of the substring must fall at the first non-*c* characters).
2. Find the length, *k*, of said substring.
3. For every substring having *k > 1*, replace the substring with a single occurrence of the character, *c*, followed by the integer value of *k*.

The above steps should be repeated until *str* no longer contains any substring of repeated characters having *k > 1*. For single occurrences of a character (*k = 1*), the character cannot be compressed and should be left as-is. For example, if *str = "aaaaabbbbbbbbbccccdeeeeeee"*, a compression operation would reduce *str* to *"a5b9c4de7"*. Observe that *d* is not repeated, so no compression number (*k*) is displayed after it.

Complete the *compress* function which takes a string, *str* as a parameter, performs a compression operation on *str*, and returns the compressed string.

Input Format
A single line containing string *str* of length *n*.

Constraints
- *0 < n < 1024*

Output Format
You are not responsible for printing anything to stdout. Your *compress* function must return the compressed string.

Sample Input 1

```
aaaaabbbbbbbbbccccpqrstuv
```

Sample Output 1

```
a5b9c4pqrstuv
```

Explanation 1:
The given string contains the following repeating substrings:
For "*aaaah*", *k=5*, so this substring is replaced with "*a5*".
For "*bbbbbbbbb*", *k=9*, so this substring is replaced with "*b9*".
For "*cccc*", *k=4*, so this substring is replaced with "*c4*".
As *p, q, r, s, t, u,* and *v* are all non-repeating characters, they are left as-is. Thus, the string returned by the *compress* function is `a5b9c4pqrstuv`.

Question - 5
Try arranging

Given an array of positive and negative numbers, arrange them in an alternate fashion such that every positive number is followed by negative and vice-versa

maintaining the order of appearance.

Number of positive and negative numbers need not be equal. If there are more positive numbers they appear at the end of the array. If there are more negative numbers, they too appear in the end of the array

Sample Input:
6
1
2
3
-4
-1
4

Output:
1
-4
2
-1
3
4

Explanation : 6 is the number of input elements {1, 2, 3, -4, -1, 4} are the array elements
Output: is the array { 1,-4,2, -1, 3, 4} with every element printed in a new line.

Question - 6

Vowels and Consonants

Write a Program to count the number of vowels and Consonants in a given word

Sample Input : How Are You
Sample Output : 5 Vowels and 4 Consonants

Sample Input: GYPSY
Sample Output: 5 Consonants

Sample Input: aeiou
Sample Output: 5 Vowels