

Problem A. Waleed's Thankfulness

Input file: **standard input**
Output file: **standard output**
Time limit: **3 seconds**
Memory limit: **64 megabytes**

Waleed opens his eyes and tries to stand up but finds his body weirdly light. When he turns to sort out his bed, he can't believe what he's seeing. His body is still on the bed! He is dead. He gets scared, tries to call his family, father, mother, brother, friends. No one can hear him. He can see them all being busy with their lives and no one cares or come to help him. He runs to them, tries to get their attentions but no one hears him. "That's it Waleed, your life is over and no one can help you." He thinks to himself. He's regretting wasting all the blessing he had and being unthankful. He starts praying to God to give him a second chance. While he is crying and begging God to forgive him, his alarm rings. It was all a dream.

After the dream, Waleed realized how blessed and fortunate he is and decided to list down 100 of the blessings in his life. Every night he check his list and writes down the blessings that he either used or was thankful for them during the day and also how many times he was unthankful. He want to make sure that he is always more thankful than unthankful so he can learn to appreciate what he has more before it's too late. To help him to know his ranking in thankfulness write a program, which would calculate his thankfulness, based on how many time he remembered any of his life blessings minus times he was unthankful.

Input

First line consists of a positive integer N ($1 \leq N \leq 100$) which is total number of thankful and unthankful moments in the Waleed's day. Next line consists of N integers ($0 \leq x_i \leq 100$). If the integer is between 1 to 100 it means Waleed was thankful for that blessing and if it is zero he was unthankful. The end of input will be denoted by a case with $N = 0$. This case should not be processed.

Output

For each test case, print a line in the format, "Case X : Y" , where X is the case number and Y is the Waleed's thankfulness for this case.

Example

standard input	standard output
5	Case 1: 1
3 4 0 0 1	Case 2: -2
4	Case 3: 3
2 0 0 0	
7	
1 2 3 4 5 0 0	
0	

Problem B. To Marry or Not To Marry

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 64 megabytes

John Doe is having a very hard time figuring out weather he should marry Jane Doe or not. The trouble is, if the bond between them is not too strong, then marrying each other will not make them "Alive as a result both will be "Forever Dead".

However, John Doe came upon an ancient scroll which describes how to measure the bond between them, and ultimately know his fate. But the scroll only explains an algorithm which John Doe doesn't know how to implement.

John Doe came all the way from his grave to seek your help in implementing the algorithm so that he can find out his fate with Jane Doe and also his dead life.

The algorithm says the following, you will be given two list containing things that John Doe and Jane Doe likes, respectively. The scroll also says that the bond between them will be stronger if they are of opposite character, meaning they like different things. If equal to or more than 50% things from John's list matches with Jane's list then the he should not marry, but if it is less than 50% then he should marry.

Help John Doe decide weather to marry or not to marry Jane Doe.

Input

The first line will contain an integer $n(1 \leq n \leq 100)$, which is the number of things John Doe likes. The next line will contain n space separated strings, describing the things. Every string is unique.

The next line will contain another integer $m(1 \leq m \leq 100)$, which is the number of things Jane Doe likes. The next line will contain m space separated strings, describing the things. Every string is unique.

Length of each strings will not exceed 26 and only consists lower-case.

Output

Print out "Alive"if John Doe should marry or else "Forever Dead"if he should not, in one line without the quotes .

Examples

standard input	standard output
3 gaming puzzle sleeping 2 chocolate sleeping	Alive
4 food programming anime travelling 4 food shopping anime travelling	Forever Dead
5 pizza burger pasta spaghetti biriyani 1 vegetables	Alive

Problem C. Nasi Lemak Stall

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 64 megabytes

Amjad, Asyraf, Ayeshah, Zarir and Samir are in the queue for a "Nasi Lemak stall"; there are no other people in the queue. The first one in the queue (Amjad) buys a Nasi Lemak, eat it and then Amjad will double himself and then will queue. Let me explain that in more details:

Queue : Samir -> Zarir -> Ayeshah -> Asyraf -> Amjad —> (Nasi Lemak Stall)

1st Nasi Lemak will be eaten by Amjad and then Amjad will be doubled and queue again. The queue will look like this:

Queue : Amjad -> Amjad -> Samir -> Zarir -> Ayeshah -> Asyraf —> (Nasi Lemak Stall)

2nd Nasi Lemak will be eaten by Asyraf and then Asyraf will be doubled and queue again. The queue will look like this:

Queue : Asyraf -> Asyraf -> Amjad -> Amjad -> Samir -> Zarir -> Ayeshah —> (Nasi Lemak Stall)

The 6th and 7th Nasi Lemak will be eaten by Amjad and then Amjad will be double to 4 Amjad(s) and then they will queue again.

Write a program that will print the name of the person who will eat the n-th Nasi Lemak.

Note that in the very beginning the queue looks like that: Amjad, Asyraf, Ayeshah, Zarir, Samir. The first person is Amjad.

Input

The data consist of a single integer N ($1 \leq N \leq 10^9$).

Output

Print the single line — the name of the person who eats the N-th Nasi Lemak. The Nasi Lemak(s) are numbered starting from 1. Please note that you should spell the names like this: "Amjad "Asyraf "Ayeshah "Zarir "Samir"(without the quotes). In that order precisely the friends are in the queue initially.

Examples

standard input	standard output
1	Amjad
6	Amjad

Problem D. FizzBuzz V2

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 megabytes

FizzBuzz is an interview question designed to filter out programmers who can't program. The problem is simple, the interviewee is required to make a program which output 1 to 100. But if the number is divisible by 3, print "Fizz" instead. If the number is divisible by 5, print "Buzz" instead. If the number is divisible by both 3 and 5, print "FizzBuzz" instead.

In Code Knights round 1, all participants managed to answer this problem relatively easily. However, they had a heads up. Plus, they might google the code and just modify it a little. Do they actually code it? To make sure, (and probably because I don't have other idea) we are going to do a retry, with some modification.

This time, you are given 4 integers a , b , c and d . You are required to print integers from a to b . If the integer is divisible by c , print "Fizz". If the integer is divisible by d , print "Buzz". If the integer is divisible by c and d , print "FizzBuzz". Note that b can be lower than a and in that case, the number should be printed in decreasing order.

Input

The input consists of a single line consisting of 4 integers a , b , c and d . ($1 \leq a, b, c, d \leq 10^5$)

Output

Print, in its own line, integers from a to b . If the integer is divisible by c , print "Fizz". If the integer is divisible by d , print "Buzz". If the integer is divisible by both c and d , print "FizzBuzz".

Example

standard input	standard output
9 16 3 5	Fizz Buzz 11 Fizz 13 14 FizzBuzz 16

Note

In the first example, the number is from 9 to 16. The number 9 and 12 is divisible by 3, so they are replaced with "Fizz". The number 10 is divisible by 5, so it is replaced with "Buzz". The number 15 is divisible by both 3 and 5, so it is replaced with "FizzBuzz".