UVa Online Judge

10924 Prime Words

A prime number is a number that has only two divisors: itself and the number one. Examples of prime numbers are: 1, 2, 3, 5, 17, 101 and 10007.

In this problem you should read a set of words, each word is composed only by letters in the range `a-z` and `A-Z`. Each letter has a specific value, the letter `a` is worth 1, letter `b` is worth 2 and so on until letter `z` that is worth 26. In the same way, letter `A` is worth 27, letter `B` is worth 28 and letter `Z` is worth 52.

You should write a program to determine if a word is a prime word or not. A word is a prime word if the sum of its letters is a prime number.

Input

The input consists of a set of words. Each word is in a line by itself and has L letters, where $1 \leq L \leq 20$. The input is terminated by enf of file (EOF).

Output

For each word you should print: '`It is a prime word.`', if the sum of the letters of the word is a prime number, otherwise you should print: '`It is not a prime word.`'.

Sample Input

```
UFRN
contest
AcM
```

Sample Output

```
It is a prime word.
It is not a prime word.
It is not a prime word.
```

UVa Online Judge

617 Nonstop Travel

Fast Phil works the late shift and leaves his company's parking lot at precisely 2:00 AM every morning. His route home is by a straight road which has one or more traffic signals. Phil has always wondered if, given the locations and cycles of each of the traffic signals, are there velocities he can travel home without ever having to speed up or slow down on account of a red light. You are to write a program to satisfy his curiosity.

Your program should find all integer speeds (in miles per hour) which can be used for Phil's trip home. Each speed is a rate (in miles per hour) he can maintain the moment he leaves the parking lot at 2:00 AM until he arrives home (we assume Phil has a long driveway in which to decelerate) such that he never passes through a red signal. He is allowed to pass throgh a signal at the exact moment it turns from yellow to red, or at the exact moment a red signal turns green. Since Phil is a relatively law-abiding citizen, you need only consider speeds less than or equal to 60 mph. Likewise, Phil isn't interested in travelling too slowly, so you should not consider speeds lower than 30 mph.

Input

Input will consist of one or more sets of data describing a set of traffic signals, followed by the integer '`-1`'. The first integer in each set will contain the value N (specifying the number of traffic signals). N will be no larger than 6. This value will be followed by N sets of numbers, each containing values (in order) for L, G, Y and R. L is a positive real number indicating the location of a traffic signal, in miles, from the parking lot. G, Y and R are the lengths of time (in seconds) of the green, yellow, and red periods of the corresponding traffic signal's cycle. Phil has learned from an informant in the Highway Department that all N traffic signals start their green period precisely at 2:00 AM.

Output

Output should consist of the input case number (starting with 1) followed by a list of all valid integer speeds Phil may drive to avoid the red signals. Consecutive integer speeds should be specified in interval notation of the form L-H, where L and H are the lowest and highest speeds for the interval. Intervals of the form L-L (that is, an interval of length 0) shold just be written as L. Intervals should be separated by commas. If there are no valid speeds, you program should display the phrase '`No acceptable speeds`'. The Expected Output below illustrates this format.

Sample Input

```
1
5.5  40 8 25

3
10.7  10 2 75
12.5  12 5 57
17.93 15 4 67

-1
```

Sample Output

```
Case 1: 30, 32-33, 36-38, 41-45, 48-54, 59-60
```

```
Case 2: No acceptable speeds.
```

UVa Online Judge

10268 498-bis

Looking throw the "Online Judge's Problem Set Archive" I found a very interesting problem number 498, titled "Polly the Polynomial". Frankly speaking, I did not solve it, but I derived from it this problem.

Everything in this problem is a derivative of something from 498. In particular, 498 was "*... designed to help you remember ... basic algebra skills, make the world a better place, etc., etc.*". This problem is designed to help you remember basic derivation algebra skills, increase the speed in which world becomes a better place, etc., etc.

In 498 you had to evaluate the values of polynomial

$$a_0x^n + a_1x^{n-1} + \ldots + a_{n-1}x + a_n.$$

In this problem you should evaluate its derivative. Remember that derivative is defined as

$$a_0nx^{n-1} + a_1(n-1)x^{n-2} + \ldots + a_{n-1}.$$

All the input and output data will fit into integer, i.e. its absolute value will be less than 2^{31}.

Input

Your program should accept an even number of lines of text. Each pair of lines will represent one problem. The first line will contain one integer - a value for x. The second line will contain a list of integers a_0, a_1, ..., a_{n-1}, a_n, which represent a set of polynomial coefficients.

Input is terminated by <EOF>.

Output

For each pair of lines, your program should evaluate the derivative of polynomial for the given value x and output it in a single line.

Sample Input

```
7
1 -1
2
1 1 1
```

Sample Output

```
1
5
```

UVa Online Judge

441 Lotto

In the German Lotto you have to select 6 numbers from the set {1,2,...,49}. A popular strategy to play Lotto - although it doesn't increase your chance of winning — is to select a subset S containing k ($k > 6$) of these 49 numbers, and then play several games with choosing numbers only from S.

For example, for $k = 8$ and $S = \{1, 2, 3, 5, 8, 13, 21, 34\}$ there are 28 possible games: [1,2,3,5,8,13], [1,2,3,5,8,21], [1,2,3,5,8,34], [1,2,3,5,13,21], ..., [3,5,8,13,21,34].

Your job is to write a program that reads in the number k and the set S and then prints all possible games choosing numbers only from S.

Input

The input file will contain one or more test cases.

Each test case consists of one line containing several integers separated from each other by spaces. The first integer on the line will be the number k ($6 < k < 13$). Then k integers, specifying the set S, will follow in ascending order.

Input will be terminated by a value of zero (0) for k.

Output

For each test case, print all possible games, each game on one line.

The numbers of each game have to be sorted in ascending order and separated from each other by exactly one space. The games themselves have to be sorted lexicographically, that means sorted by the lowest number first, then by the second lowest and so on, as demonstrated in the sample output below.

The test cases have to be separated from each other by exactly one blank line. Do not put a blank line after the last test case.

Sample Input

```
7 1 2 3 4 5 6 7
8 1 2 3 5 8 13 21 34
0
```

Sample Output

```
1 2 3 4 5 6
1 2 3 4 5 7
1 2 3 4 6 7
1 2 3 5 6 7
1 2 4 5 6 7
1 3 4 5 6 7
2 3 4 5 6 7

1 2 3 5 8 13
1 2 3 5 8 21
1 2 3 5 8 34
1 2 3 5 13 21
1 2 3 5 13 34
```

```
1 2 3 5 21 34
1 2 3 8 13 21
1 2 3 8 13 34
1 2 3 8 21 34
1 2 3 13 21 34
1 2 5 8 13 21
1 2 5 8 13 34
1 2 5 8 21 34
1 2 5 13 21 34
1 2 8 13 21 34
1 3 5 8 13 21
1 3 5 8 13 34
1 3 5 8 21 34
1 3 5 13 21 34
1 3 8 13 21 34
1 5 8 13 21 34
2 3 5 8 13 21
2 3 5 8 13 34
2 3 5 8 21 34
2 3 5 13 21 34
2 3 8 13 21 34
2 5 8 13 21 34
3 5 8 13 21 34
```

UVa Online Judge

11614 Etruscan Warriors Never Play Chess

A troop of Etruscan warriors is organized as follows. In the first row, there is only one warrior; then, the second row contains two warriors; the third row contains three warriors, and so on. In general, each row i contains i warriors.

We know the number of Etruscan warriors of a given troop. You have to compute the number of rows in which they are organized.

Please note that there may be some remaining warriors (this could happen if they are not enough to form the next row). For example, 3 warriors are organized in 2 rows. With 6 warriors you can form 3 rows; but you can also form 3 rows with 7, 8 or 9 warriors.

Input

The first line of the input contains an integer indicating the number of test cases.

For each test case, there is a single integer, n, indicating the number of Etruscan warriors. You can assume that $0 \leq n \leq 10^{18}$.

Output

For each test case, the output should contain a single integer indicating the number of rows that can be formed.

Sample Input

```
6
3
6
7
8
9
10
```

Sample Output

```
2
3
3
3
3
4
```

UVa Online Judge

10101 Bangla Numbers

Bangla numbers normally use 'kuti' (10000000), 'lakh' (100000), 'hajar' (1000), 'shata' (100) while expanding and converting to text. You are going to write a program to convert a given number to text with them.

Input

The input file may contain several test cases. Each case will contain a non-negative number ≤ 999999999999999.

Output

For each case of input, you have to output a line starting with the case number with four digits adjustment followed by the converted text.

Sample Input

```
23764
45897458973958
```

Sample Output

```
1. 23 hajar 7 shata 64
2. 45 lakh 89 hajar 7 shata 45 kuti 89 lakh 73 hajar 9 shata 58
```