ACM ICPC World Finals 2013
Warmup Contest at UVa Site

Universidad de Valladolid

Bangladesh Association of Problem Setters

Hosted by UVa Online Judge, Spain

25th May 2013
You get 15 Pages
10 Problems
&
300 Minutes

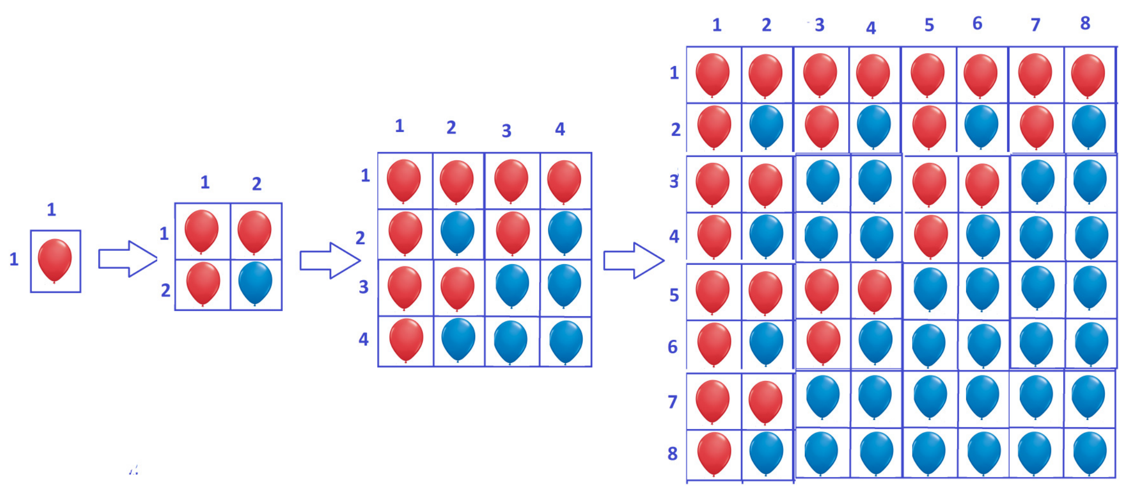
acm International Collegiate Programming Contest

IBM event sponsor

A	**Erratic Expansion** **Input:** Standard Input **Output:** Standard Output	

Piotr found a magical box in heaven. Its magic power is that if you place any red balloon inside it then, after one hour, it will multiply to form 3 red and 1 blue colored balloons. Then in the next hour, each of the red balloons will multiply in the same fashion, but the blue one will multiply to form 4 blue balloons. This trend will continue indefinitely.

The arrangements of the balloons after the 0^{th}, 1^{st}, 2^{nd} and 3^{rd} hour are depicted in the following diagram.



As you can see, a red balloon in the cell (i, j) (that is i^{th} row and j^{th} column) will multiply to produce 3 red balloons in the cells (i*2 - 1, j*2 - 1), (i*2 - 1, j*2), (i*2, j*2 - 1) and a blue balloon in the cell (i*2, j*2). Whereas, a blue balloon in the cell (i, j) will multiply to produce 4 blue balloons in the cells (i*2 - 1, j*2 - 1), (i*2 - 1, j*2), (i*2, j*2 - 1) and (i*2, j*2). The grid size doubles (in both the direction) after every hour in order to accommodate the extra balloons.

In this problem, Piotr is only interested in the count of the red balloons; more specifically, he would like to know the total number of red balloons in all the rows from **A** to **B** after $\mathbf{K^{th}}$ hour.

Input

The first line of input is an integer **T**(**T**<1000) that indicates the number of test cases. Each case contains 3 integers **K**, **A** and **B**. The meanings of these variables are mentioned above. **K** will be in the range **[0, 30]** and $1 \leq \mathbf{A} \leq \mathbf{B} \leq 2^{\mathbf{K}}$.

Output

For each case, output the case number followed by the total number of red balloons in rows [**A**, **B**] after K^{th} hour.

Sample Input

```
3
0 1 1
3 1 8
3 3 7
```

Output for Sample Input

```
Case 1: 1
Case 2: 27
Case 3: 14
```

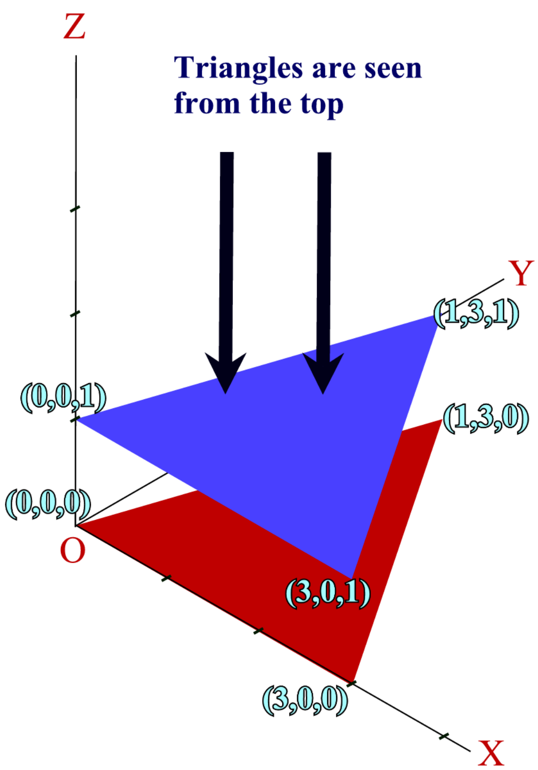
Problemsetter: Sohel Hafiz, Special Thanks: Md. Mahbubul Hasan

B — Two Triangles in 3D

Input: Standard Input
Output: Standard Output

There is a red triangle and a blue triangle in 3D space.

When looking from the top (from +z to XY plane), how much red and how much blue would be seen? (Assume you are seeing both the triangles clearly from an infinite distance)



The triangles are opaque so they can block each other.

For example, if the red one is (0,0,0),(3,0,0) and (1,3,0), the blue one is (0,0,1),(3,0,1) and (1,3,1) (See the figure on the left). When looking from the top, both becomes (0,0), (3,0), (1,3), but you can only see the blue one. This figure corresponds to the 3rd sample input.

Input

The first line contains the number of test cases $T(1 \le T \le 3000)$.

Each test case contains two lines.

Each line contains 9 integers to describe a triangle: x_1 y_1 z_1 x_2 y_2 z_2 x_3 y_3 z_3 $(0 \le x_i, y_i, z_i \le 100)$. The first line describes the red one, and the second line the blue one. Each triangle is guaranteed to have a strictly positive area. They would never lie on the same plane (As in that case you can't tell which one blocks the other).

Output

For each test case, print two real numbers, the red area and the blue area that can be seen, rounded to 4 decimal places. Absolute errors less than $1.1*10^{-4}$ will definitely be ignored.

Sample Input

```
3
0 0 0 1 0 0 0 1 0
0 0 1 1 0 1 0 1 1
0 0 0 1 0 1 0 1 0
1 0 0 0 0 1 1 1 0
0 0 0 3 0 0 1 3 0
0 0 1 3 0 1 1 3 1
```

Output for Sample Input

```
0.0000 0.5000
0.3750 0.3750
0.0000 4.5000
```

Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan, Shahriar Manzoor

Explanation of sample:

In the first and third case, the red triangle is completely blocked by the blue one.

In the second case, two triangles intersect, each is divided into two parts.

C	# Rectangle XOR Game **Input:** Standard Input **Output:** Standard Output	acm

There is a 01 matrix with n rows and m columns, having at least one 1. Rows are numbered 1 to n from top to bottom, columns are numbered 1 to m from left to right.

Two players move in turn. In each move, the player selects a rectangle (x_1,y_1)-(x_2,y_2) ($1 \le x_1 \le x_2 \le n$, $1 \le y_1 \le y_2 \le m$) and flips all the numbers ($0 \to 1$, $1 \to 0$) in it (i.e. the flip all positions (x,y) where $x_1 \le x \le x_2$, $y_1 \le y \le y_2$)). The only restriction is that the top-left corner (i.e. (x_1,y_1)) must be changing from 1 to 0.

After a player's move, if the matrix contains only zeros, he wins.

Your task is to determine whether the first player can win, if both players play perfectly.

If he can win, count how many different first moves can lead to victory, and print the lexicographically smallest one (i.e. x_1 should be as smallest as possible, then y_1, x_2, y_2)

Input

The input contains at most 400 test cases. Each case begins with two integers n and m ($1 \le n, m \le 100$). The next n lines contain the matrix. Each line contains m integers (Either 0 or 1). There will be at least one 1 in the matrix.

The input ends with EOF

Output

For each test case, if the first player cannot win, print a single string "No" (without quotes), otherwise print five integers c x_1 y_1 x_2 y_2, where c is the number of winning first moves, and (x_1,y_1)-(x_2,y_2) is the lexicographically smallest first move.

Sample Input	Output for Sample Input
2 2 0 1 1 0 2 2 1 0 0 1 2 2 1 1 1 1 1 7 0 1 1 1 1 0 0	No 1 1 1 2 2 3 1 1 2 2 3 1 2 1 5

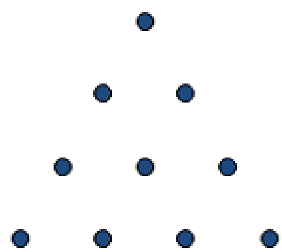
Problemsetter: Rujia Liu, Special Thanks: Md. Mahbubul Hasan, Yubin Wang

Explanation below:

```
Case 1:

2 2
0 1
1 0

The first player loses, because he has 4 moves, leading to:

0 0
1 0

0 0
1 1

0 1
0 0

0 1
0 1

The second player can flip all 1s to 0s.

Case 2:

2 2
1 0
0 1

This is winning, because the first player can flip the whole board and turn to the
losing game analyzed above.


Case 3:

2 2
1 1
1 1

There are 3 winning moves, leading to the following games:

0 0
0 0

1 0
1 1

1 1
0 1

The lexicographically first one is (1,1)-(2,2)
```

D	# Equilateral Triangle in a Grid **Input:** Standard Input **Output:** Standard Output

An equilateral triangle with an integer number of n equidistantly distributed 'dots' along each side, forms a 'triangle grid' of uniformly distributed 'dots' within the boundary of this equilateral triangle. The figure below illustrates such a 'triangle grid' for the example $n = 4$.



Question:

How many different equilateral triangles can be found in this 'triangle grid' when the vertexes of each equilateral triangle must always be 'dots' of the 'triangle grid'?

Note:

Take into account that the equilateral triangles come with multiple sizes and orientations!

Input

Each line of the input file contains a single integer n denoting the number of 'dots' along each side of the equilateral triangle ($2 \leq n \leq 100$). The input is terminated by a line containing a single zero. Input file will contain less than 20 lines.

Output

For each line of input produce one line of output. This line should contain the number of different equilateral triangles that can be found in the 'triangle grid' of the size specified by the according input line.

Sample Input	Output for Sample Input
2 3 0	1 5

Problemsetter: Eric Schmidt, Special Thanks: Shahriar Manzoor

E	Chocolate or Money **Input:** Standard Input **Output:** Standard Output

Nowadays it is a trend to give one chocolate instead of 1 taka (currency of Bangladesh) and even two chocolates instead of 2 taka. Though both kids and shopkeeper may be happy with these, but not everyone. I was thinking what if we buy a packet of 50 chocolates by 40 taka and use these instead of money. May be while buying a car instead of giving a check of 1,000,000 taka we can give 20,000 packets of chocolates (and thus saving 200,000 taka)!! How weird thinking! But then I thought if one does not have 1 taka coin then yes one chocolate can save my day! But do we need chocolate always? Or better to ask, if we don't have chocolate then can other items save our day? Let's go and check it.

Instead of using chocolates, let's say there are N types of items that can be exchanged. Suppose P(x) = number of ways a subset of the given items can be chosen so that if the buyer and the seller both have sufficient number of these items (in subset) they can exchange x taka.

For example, there are three items, X, Y and Z and their prices are 2, 4, and 2 taka respectively. Then P(1) = 0, P(2) = 6 (X, Z, ZY, XZ, YZ, XYZ), P(3) = 0, P(4) = 7 (X, Y, Z, XY, YZ, XZ, XYZ).

And you will also be given an integer C, you have to find P(1) + ... + P(C) or $\sum_{x=1}^{C} P(x)$. So, if C = 4, then for the above case, the result is 0 + 6 + 0 + 7 = 13.

However, it is not always possible to have N items in a shop right? So we are also interested to find out the sum if we give restriction that there is exactly K items in the shop (instead of any item). That is, the size of the subset is exactly K. Say for K = 2, for the above case, P(1) = 0, P(2) = 3 (XY, YZ, ZX), P(3) = 0, P(4) = 3 (XY, YZ, ZX). So answer is: 6.

Input

In the input file, first line contains number of test cases, T ($T \leq 40$). Hence followed by three positive numbers N, C and K. ($N \leq 10^5$, $C \leq 10^{15}$, $K \leq N$). In the next line, there are 3 integers a, b and c ($0 < a, b, c \leq 10^5$). With the help of these 3 integers you can generate the price of all N things. It can be generated as follows:

Price[0] = a
Price[1] = b
Price[i] = 1 + (a*Price[i-2] + b*Price[i-1] + c)%100000 (for $2 \leq i < N$)

Output

For each case, output the number of test case and then the required solutions, first one is any size subset and second one is for K size subset. Since the answers can be very big print them in modulo 1,000,000,007. For details please follow the sample input output.

Sample Input	Output for Sample Input
1 3 4 2 2 4 2	Case 1: 17 10

Problemsetter: Md. Mahbubul Hasan, Special Thanks: Hasnain Heickal, Jane Alam Jan

F	# Sum of Totatives **Input:** Standard Input **Output:** Standard Output	

When Tomisu's student Hedayet came to him with an open problem which he (Hedayet) thought was true after running a lengthy simulation, Tomisu's face lit up with a sense of glory. But soon his face dimmed because he was able to prove it within one day and the proof was a mere four liner (Such an easy and obvious proof will not bring glory). So now with a gloomy face he begins to write a contest problem (What else can he do after all?) which came to his mind as a bi-product of proving that theorem. Although as dumb as he is, he does not have any clue on how to solve this new problem, so he plans to ask Snapdragon for help. But before that he needs to write a crystal clear statement :). You may also help Tomisu solve this problem.

Let $\mathbf{a \geq 2}$ be a positive integer, then $\boldsymbol{\varphi}\mathbf{(a)}$ denotes the number of "totatives" of a, i.e. those positive integers $\mathbf{r < a}$ such that $r \perp a$ (in other word ***gcd(r,a)*** equals **1**). For example $\boldsymbol{\varphi}\mathbf{(25) = 20}$. There is another less known function titled "sum of totatives" which as the name indicates is the summation of these totatives. The symbol for this function is $\boldsymbol{\varphi_1}$. For example $\boldsymbol{\varphi_1}\mathbf{(25)}$ = 1 + 2 + 3 + 4 + 6 + 7 + 8 + 9 + 11 + 12 + 13 + 14 + 16 + 17 + 18 + 19 + 21 + 22 + 23 + 24 = 250.

Now lets consider an equation $\boldsymbol{\varphi_1}\mathbf{(I) = M*}\boldsymbol{\varphi_1}\mathbf{(J)}$, here $\mathbf{I \neq J}$ and **M** is also a positive integer. Now given the value of **M**, you will have to find a possible value for **I**, **J** such that $\boldsymbol{\varphi_1}\mathbf{(I) = M*}\boldsymbol{\varphi_1}\mathbf{(J)}$.

Input

The input file contains at most **400** lines of inputs. Each line contains a single positive integer which denotes the value of **M**. The value of **M** does not exceed $\mathbf{10^9}$. For at least **80%** cases the value of **M** does not even exceed $\mathbf{10^8}$. Input is terminated by a line containing a single zero.

Output

For each set of input produce one line of output. This line contains three integers which denote the value of **M**, **I** and **J** respectively. The value of **M** will be such that either there will be possible values of **I** and **J** not exceeding $\mathbf{10^{16}}$ or there will be no possible value of **I** and **J**. For the latter case print the line “No solution” (without the quotes) instead the value of **I** and **J**, but you need to print the value of **M** before that. You should make sure that $\mathbf{(2 \leq I, J \leq 10^{16})}$.

Sample Input	Output for Sample Input
3 7 0	3 3 2 7 7 3

Problemsetter: Shahriar Manzoor, Moderator and alternate writer: Derek Kisman
Special Thanks: Mohammad Hedayet

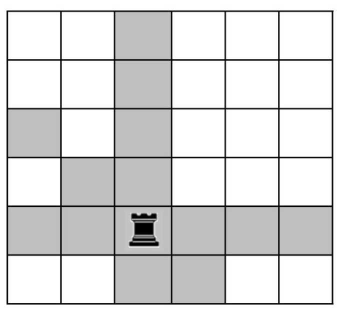
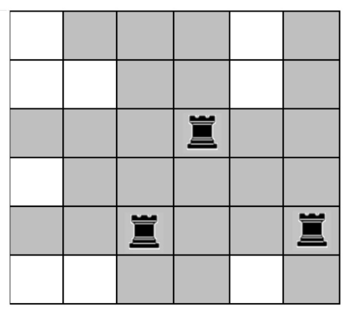
G Super Rooks on a Chessboard

Input: Standard Input
Output: Standard Output

Let's assume there is a new chess piece named Super-rook. When placed at a cell of a chessboard, it attacks all the cells that belong to the **same row** or **same column**. Additionally it attacks all the cells of **the diagonal** that goes from **top-left** to **bottom-right** direction through that cell.

N Super-rooks are placed on a **R x C** chessboard. The rows are numbered **1** to **R** from top to bottom and columns are numbered **1** to **C** from left to right of the chessboard. You have to find the number of cells of the chessboard which are **not attacked** by any of the Super-rooks.

The picture on the left shows the attacked cells when a Super-rook is placed at cell (5, 3) of a 6 x 6 chessboard. And the picture on the right shows the attacked cells when three Super-rooks are placed at cells (3, 4) , (5, 3) and (5, 6). These pictures (Left and right one) corresponds to the first and second sample input respectively.

Input
First line of input contains an integer **T(1 ≤ T ≤ 20)** which is the number of test cases. The first line of each test case contains three integers **R**, **C** and **N(1 ≤ R, C, N ≤ 50,000)**. The next **N** lines contain two integers **r, c** giving the row and column of a Super-rook on the chessboard (**1 ≤ r ≤ R** and **1 ≤ c ≤ C**). You may assume that two Super-rooks won't be placed on the same cell.

Output
For each test case, output the case number followed by the number of cells which are **not attacked** by any of the Super-rook.

Sample Input

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

Output for Sample Input

```
Case 1: 22
Case 2: 9
```

Problemsetter: Tasnim Imran Sunny, Special Thanks: Md. Mahbubul Hasan

H	Pairing Boys and Girls **Input:** Standard Input **Output:** Standard Output	acm

In a dance party, Boys and Girls make a line before a song starts. Then a boy will pair up with a girl who is somewhere later in the line. The pairing will occur in such a way that **number of pairs** get **maximized**.

For example: If the Boy-Girl sequence in a line looks like: GGBGGBGB then there can be 2 pairs maximum. If we consider the line to be 1 indexed, the boy standing at 3^{rd} position pairs up with the girl at 4^{th} or 5^{th} position and the boy at 6^{th} position will pair up with the girl at 7^{th} position. They boy at 8^{th} position can not propose any girl, since there is no girl after him. Again, the boy at 3^{rd} position could not pair up with the girl at 7^{th} position, because if he would have paired up with her the number of pairing would be only 1, which is not maximum possible pairing.

After each song, boys and girls would line up in the same way before a new song. In this time, bunch of boys or girls may come. Manager will insert them at a certain position in the line. Then again pairing will occur. After each insertion you have to print number of maximum possible pairings. Initially there would be a boy in the line.

Input

First line of the input file will contain a positive integer **T** (T ≤ **10**). Hence follow **T** test cases.

In the first line of each test case there will be a positive integer **N** (**N** ≤ **100,000**). Each of the next **N** lines describes an insertion operation for a group. Each line will contain **3** integers "***type where count***".

- ***type*** is **0** if the group is of **boy**s, ***type*** will be **1** if the group is of **girl**.
- ***where*** denotes the position where this group will be inserted. If ***where*** = **0**, then this group will be inserted in front of the entire line, otherwise that group will be inserted after ***where*** number of people. It will not exceed total number people in the line.
- ***count*** denotes number of boys/girls in that group. **1** ≤ ***count*** ≤ **10,000**

Output

For each case output the case number "`Case x:`". Hence for each insertion operation, output the maximum possible pairing after the insertion operation is performed. There will be **no blank line** between output for different test cases. For details please go through the sample input output.

Sample Input	Output for Sample Input
2 3 1 1 4 0 3 3 0 0 2 1 1 0 1	Case 1: 1 3 4 Case 2: 0

Problemsetter: Md. Mahbubul Hasan, Special Thanks: Tasnim Imran Sunny

I	**Bisection Method** **Input:** Standard Input **Output:** Standard Output	

Bisection method is a very basic and robust numerical method for finding roots of an equation. Finding the roots of a nonlinear equation which **f(x)**=0 is equivalent to finding the values of x for which **f(x)** is zero or approximately zero. In bisection method to find the roots of an equation we first need two initial guesses x_l and x_u which bracket a root (Or more than one root), that means $f(x_l)f(x_u) < 0$. This ensures that the function must become zero somewhere in between and so it is guaranteed that there is at least one root between $\mathbf{x_l}$ and $\mathbf{x_u}$. The bisection algorithm works the following way:

1. Choose x_l and x_u such that $f(x_l)f(x_u) < 0$ and $x_l < x_u$

2. Estimate the approximate root $x_r = \dfrac{x_l + x_u}{2}$

3. $if(f(x_l)f(x_r) < 0)$ set $x_u = x_r$
   $if(f(x_l)f(x_r) > 0)$ set $x_l = x_r$
   $if(f(x_l)f(x_r) = 0)$ set x_r is the root

4. If root is not found go back to 2.

In this problem your job is not to find the roots of a function f(x) using bisection method. In this problem you will be given an equation of the form $(x-r_1)(x-r_2)(x-r_3)... (x-r_n)=0$, so it is obvious that the roots of this equation are $r_1, r_2, r_3,..., r_n$. For this problem all the roots are strictly positive integers less than 10000 and the range of x_l and x_u is $0 \le \mathbf{x_l} < \mathbf{x_u} \le 10000$. Now your job is to find that for a given root, how many possible pairings of (x_l, x_u) are there for which that root is found in at most 7 steps?

Input

First line of the input file contains a positive integer N ($1 \le N \le 30$) which denotes how many sets of inputs are there. Each set of input consists of two lines. The description of the two lines are given below:

The first line of each set consists of an equation of the form $(x-r_1)(x-r_2)(x-r_3)... (x-r_n)=0$. Here $r_1, r_2, r_3,..., r_n$ are all integers, $0 < r_1, r_2, r_3,..., r_n < 10000$ and $0 < n < 11$. The second line contains an integer r, whose value is equal to any one of the roots.

Output

For each set of input produce one line of output. This line contains an integer which denotes of all the pairings of possible values for which root r will be found using bisection method in seven steps or less. Note that as the possible values for xl and xu is in the range from 0 to 10000. So possible pairings xi and xu are (0, 1), (0, 2), (0, 3), ...,(0, 10000), (1, 2), (1, 3), (1, 4), ..., (1, 10000), ..., (9999,10000). So total number of pairings are (10001)(10001-1)/2. Of which only small number of pairings will ensure that root r is found within 7 iterations.

Sample Input	Output for Sample Input
2 (x-8469)(x-6335)=0 8469 (x-2384)(x-7423)(x-8718)=0 8718	8930 6530

Problemsetter: Shahriar Manzoor, Special Thanks: Md. Mahbubul Hasan

J

Disguised Giveaway

Input: Standard Input
Output: Standard Output

Well, I was planning to set a problem for beginners, that is: given n distinct integers each between 2 and 10^9, find the multiplication of all pairs. For example, n = 4 and the integers are 2, 5, 8 and 6, then the result would be 10 12 16 30 48 40. As the results can be printed in any order so, I wrote a special judge for this problem.

But it was a long time ago. Now I want to finish this problem and found that I only have the answer file of the problem. The input file is missing. All you have to do is to generate the input file from the answer file.

Input

Input starts with an integer T ($T \leq 25$), denoting the number of test cases.

Each case starts with an integer n ($3 \leq n \leq 200$). The next line contains n*(n-1)/2 integers showing the all pair multiplications.

Output

For each case, print the case number and the input integers in ascending order, separated by a single space. If there are multiple solutions, print the one with the smallest first integer (after sorting), in case of tie, the one with the smallest second integer and so on. You can assume for the given input there will always be a possible solution.

Sample Input	Output for Sample Input
2 4 10 12 16 30 48 40 3 10 20 8	Case 1: 2 5 6 8 Case 2: 2 4 5

Problemsetter: Jane Alam Jan, Special Thanks: Md. Mahbubul Hasan, Rujia Liu