

Codeforces Round #325 (Div. 2)**A. Alena's Schedule**

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alena has successfully passed the entrance exams to the university and is now looking forward to start studying.

One two-hour lesson at the Russian university is traditionally called a *pair*, it lasts for two academic hours (an academic hour is equal to 45 minutes).

The University works in such a way that every day it holds exactly n lessons. Depending on the schedule of a particular group of students, on a given day, some pairs may actually contain classes, but some may be empty (such pairs are called breaks).

The official website of the university has already published the schedule for tomorrow for Alena's group. Thus, for each of the n pairs she knows if there will be a class at that time or not.

Alena's House is far from the university, so if there are breaks, she doesn't always go home. Alena has time to go home only if the break consists of at least two free pairs in a row, otherwise she waits for the next pair at the university.

Of course, Alena does not want to be sleepy during pairs, so she will sleep as long as possible, and will only come to the first pair that is presented in her schedule. Similarly, if there are no more pairs, then Alena immediately goes home.

Alena appreciates the time spent at home, so she always goes home when it is possible, and returns to the university only at the beginning of the next pair. Help Alena determine for how many pairs she will stay at the university. Note that during some pairs Alena may be at the university waiting for the upcoming pair.

Input

The first line of the input contains a positive integer n ($1 \leq n \leq 100$) — the number of lessons at the university.

The second line contains n numbers a_i ($0 \leq a_i \leq 1$). Number a_i equals 0, if Alena doesn't have the i -th pairs, otherwise it is equal to 1. Numbers a_1, a_2, \dots, a_n are separated by spaces.

Output

Print a single number — the number of pairs during which Alena stays at the university.

Examples

| |
|--------------------|
| input |
| 5 0 1 0 1 1 |
| output |
| 4 |
| input |
| 7 1 0 1 0 0 1 0 |
| output |
| 4 |
| input |
| 1 0 |
| output |
| 0 |

Note

In the first sample Alena stays at the university from the second to the fifth pair, inclusive, during the third pair she will be at the university waiting for the next pair.

In the last sample Alena doesn't have a single pair, so she spends all the time at home.

B. Laurynty and Shop

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

A little boy Laurynty has been playing his favourite game NotA for quite a while and is now very hungry. The boy wants to make sausage and cheese sandwiches, but first, he needs to buy a sausage and some cheese.

The town where Laurynty lives in is not large. The houses in it are located in two rows, n houses in each row. Laurynty lives in the very last house of the second row. The only shop in town is placed in the first house of the first row.

The first and second rows are separated with the main avenue of the city. The adjacent houses of one row are separated by streets.

Each crosswalk of a street or an avenue has some traffic lights. In order to cross the street, you need to press a button on the traffic light, wait for a while for the green light and cross the street. Different traffic lights can have different waiting time.

The traffic light on the crosswalk from the j -th house of the i -th row to the $(j + 1)$ -th house of the same row has waiting time equal to a_{ij} ($1 \leq i \leq 2$, $1 \leq j \leq n - 1$). For the traffic light on the crossing from the j -th house of one row to the j -th house of another row the waiting time equals b_j ($1 \leq j \leq n$). The city doesn't have any other crossings.

The boy wants to get to the store, buy the products and go back. The main avenue of the city is wide enough, so the boy wants to cross it **exactly once** on the way to the store and **exactly once** on the way back home. The boy would get bored if he had to walk the same way again, so he wants the way home to be different from the way to the store in at least one crossing.



Figure to the first sample.

Help Laurynty determine the minimum total time he needs to wait at the crossroads.

Input

The first line of the input contains integer n ($2 \leq n \leq 50$) — the number of houses in each row.

Each of the next two lines contains $n - 1$ space-separated integer — values a_{ij} ($1 \leq a_{ij} \leq 100$).

The last line contains n space-separated integers b_j ($1 \leq b_j \leq 100$).

Output

Print a single integer — the least total time Laurynty needs to wait at the crossroads, given that he crosses the avenue only once both on his way to the store and on his way back home.

Examples

| input |
|--------------------------------|
| 4 1 2 3 3 2 1 3 2 2 3 |
| output |
| 12 |

| input |
|--------------------------|
| 3 1 2 3 3 2 1 3 |
| output |
| 11 |

| input |
|--------------------|
| 2 1 1 1 1 |
| output |
| 4 |

Note

The first sample is shown on the figure above.

In the second sample, Laurynty's path can look as follows:

- Laurenty crosses the avenue, the waiting time is **3**;
- Laurenty uses the second crossing in the first row, the waiting time is **2**;
- Laurenty uses the first crossing in the first row, the waiting time is **1**;
- Laurenty uses the first crossing in the first row, the waiting time is **1**;
- Laurenty crosses the avenue, the waiting time is **1**;
- Laurenty uses the second crossing in the second row, the waiting time is **3**.

In total we get that the answer equals **11**.

In the last sample Laurenty visits all the crossings, so the answer is **4**.

C. Gennady the Dentist

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Gennady is one of the best child dentists in Berland. Today n children got an appointment with him, they lined up in front of his office.

All children love to cry loudly at the reception at the dentist. We enumerate the children with integers from 1 to n in the order they go in the line. Every child is associated with the value of his *confidence* p_i . The children take turns one after another to come into the office; each time the child that is the first in the line goes to the doctor.

While Gennady treats the teeth of the i -th child, the child is crying with the volume of v_i . At that the *confidence* of the first child in the line is reduced by the amount of v_i , the second one — by value $v_i - 1$, and so on. The children in the queue after the v_i -th child almost do not hear the crying, so their *confidence* remains unchanged.

If at any point in time the *confidence* of the j -th child is less than zero, he begins to cry with the volume of d_j and leaves the line, running towards the exit, without going to the doctor's office. At this the *confidence* of all the children after the j -th one in the line is reduced by the amount of d_j .

All these events occur immediately one after the other in some order. Some cries may lead to other cries, causing a chain reaction. Once in the hallway it is quiet, the child, who is first in the line, goes into the doctor's office.

Help Gennady the Dentist to determine the numbers of kids, whose teeth he will cure. Print their numbers in the chronological order.

Input

The first line of the input contains a positive integer n ($1 \leq n \leq 4000$) — the number of kids in the line.

Next n lines contain three integers each v_i, d_i, p_i ($1 \leq v_i, d_i, p_i \leq 10^6$) — the volume of the cry in the doctor's office, the volume of the cry in the hall and the *confidence* of the i -th child.

Output

In the first line print number k — the number of children whose teeth Gennady will cure.

In the second line print k integers — the numbers of the children who will make it to the end of the line in the increasing order.

Examples

| input |
|--|
| 5 4 2 2 4 1 2 5 2 4 3 3 5 5 1 2 |
| output |
| 2 1 3 |

| input |
|--|
| 5 4 5 1 5 3 9 4 1 2 2 1 8 4 1 9 |
| output |
| 4 1 2 4 5 |

Note

In the first example, Gennady first treats the teeth of the first child who will cry with volume 4 . The confidences of the remaining children will get equal to $-2, 1, 3, 1$, respectively. Thus, the second child also cries at the volume of 1 and run to the exit. The confidence of the remaining children will be equal to $0, 2, 0$. Then the third child will go to the office, and cry with volume 5 . The other children won't bear this, and with a loud cry they will run to the exit.

In the second sample, first the first child goes into the office, he will cry with volume 4 . The confidence of the remaining children will be equal to $5, -1, 6, 8$. Thus, the third child will cry with the volume of 1 and run to the exit. The confidence of the remaining children will be equal to $5, 5, 7$. After that, the second child goes to the office and cry with the volume of 5 . The confidences of the remaining children will be equal to $0, 3$. Then the fourth child will go into the office and cry with the volume of 2 . Because of this

the confidence of the fifth child will be 1, and he will go into the office last.

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The protagonist of the game Philip is located in one end of the tunnel and wants to get out of the other one. The tunnel is a rectangular field consisting of three rows and n columns. At the beginning of the game the hero is in some cell of the leftmost column. Some number of trains rides towards the hero. Each train consists of two or more neighbouring cells in some row of the field.

Your task is to answer the question whether there is a sequence of movements of Philip, such that he would be able to get to the rightmost column.



Then follows the description of t sets of the input data.

Output

For each set of the input data print on a single line word YES, if it is possible to win the game and word NO otherwise.

| input | |
|------------------|--|
| 2 | |
| 16 4 | |
| ...AAAAA..... | |
| s.BBB.....CCCCC | |
|DDDDD... | |
| 16 4 | |
| ...AAAAA..... | |
| s.BBB....CCCCC.. | |
|DDDDD.... | |
| output | |
| YES | |
| NO | |

| input |
|--|
| 2 10 4 s.ZZ.....AAABB .YYYYYY... 10 4 s.ZZ.....AAAABB .YYYYYY... |
| output |
| YES NO |

In the first set of the input of the first sample Philip must first go forward and go down to the third row of the field, then go only forward, then go forward and climb to the second row, go forward again and go up to the first row. After that way no train blocks Philip's path, so he can go straight to the end of the tunnel.

Note that in this problem the challenges are restricted to tests that contain only one testset.

E. Alice, Bob, Oranges and Apples

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice and Bob decided to eat some fruit. In the kitchen they found a large bag of oranges and apples. Alice immediately took an orange for herself, Bob took an apple. To make the process of sharing the remaining fruit more fun, the friends decided to play a game. They put multiple cards and on each one they wrote a letter, either 'A', or the letter 'B'. Then they began to remove the cards one by one from left to right, every time they removed a card with the letter 'A', Alice gave Bob all the fruits she had at that moment and took out of the bag as many apples and as many oranges as she had before. Thus the number of oranges and apples Alice had, did not change. If the card had written letter 'B', then Bob did the same, that is, he gave Alice all the fruit that he had, and took from the bag the same set of fruit. After the last card was removed, all the fruit in the bag were over.

You know how many oranges and apples was in the bag at first. Your task is to find any sequence of cards that Alice and Bob could have played with.

Input

The first line of the input contains two integers, x, y ($1 \leq x, y \leq 10^{18}, xy > 1$) — the number of oranges and apples that were initially in the bag.

Output

Print any sequence of cards that would meet the problem conditions as a *compressed* string of characters 'A' and 'B'. That means that you need to replace the segments of identical consecutive characters by the number of repetitions of the characters and the actual character. For example, string AAABAABBB should be replaced by string 3A1B2A3B, but cannot be replaced by 2A1A1B2A3B or by 3AB2A3B. See the samples for clarifications of the output format. The string that you print should consist of at most 10^6 characters. It is guaranteed that if the answer exists, its compressed representation exists, consisting of at most 10^6 characters. If there are several possible answers, you are allowed to print any of them.

If the sequence of cards that meet the problem statement does not exist, print a single word Impossible.

Examples

| |
|---------------|
| input |
| 1 4 |
| output |
| 3B |

| |
|---------------|
| input |
| 2 2 |
| output |
| Impossible |

| |
|---------------|
| input |
| 3 2 |
| output |
| 1A1B |

Note

In the first sample, if the row contained three cards with letter 'B', then Bob should give one apple to Alice three times. So, in the end of the game Alice has one orange and three apples, and Bob has one apple, in total it is one orange and four apples.

In second sample, there is no answer since one card is not enough for game to finish, and two cards will produce at least three apples or three oranges.

In the third sample, cards contain letters 'AB', so after removing the first card Bob has one orange and one apple, and after removal of second card Alice has two oranges and one apple. So, in total it is three oranges and two apples.

F. Lizard Era: Beginning

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In the game Lizard Era: Beginning the protagonist will travel with three companions: Lynn, Meliana and Worrigan. Overall the game has n mandatory quests. To perform each of them, you need to take **exactly two** companions.

The attitude of each of the companions to the hero is an integer. Initially, the attitude of each of them to the hero is neutral and equal to 0. As the hero completes quests, he makes actions that change the attitude of the companions, whom he took to perform this task, in positive or negative direction.

Tell us what companions the hero needs to choose to make their attitude equal after completing all the quests. If this can be done in several ways, choose the one in which the value of resulting attitude is greatest possible.

Input

The first line contains positive integer n ($1 \leq n \leq 25$) — the number of important tasks.

Next n lines contain the descriptions of the tasks — the i -th line contains three integers l_i, m_i, w_i — the values by which the attitude of Lynn, Meliana and Worrigan respectively will change towards the hero if the hero takes them on the i -th task. All the numbers in the input are integers and do not exceed 10^7 in absolute value.

Output

If there is no solution, print in the first line "Impossible".

Otherwise, print n lines, two characters is each line — in the i -th line print the first letters of the companions' names that hero should take to complete the i -th task ('L' for Lynn, 'M' for Meliana, 'W' for Worrigan). Print the letters in any order, if there are multiple solutions, print any of them.

Examples

| |
|------------------------------|
| input |
| 3 1 0 0 0 1 0 0 0 1 |
| output |
| LM MW MW |

| |
|--|
| input |
| 7 0 8 9 5 9 -2 6 -8 -7 9 4 5 -4 -9 9 -4 5 2 -6 8 -7 |
| output |
| LM MW LM LW MW LM LW |

| |
|---------------------|
| input |
| 2 1 0 0 1 1 0 |
| output |
| Impossible |

