НБУ
НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
София 1635, ул. Монтевидео 21
тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика

Школа „Състезателно програмиране”

СЪСТЕЗАНИЕ, 23 март 2013 г.

A. Trick or Treat

Johnny and his friends have decided to spend Halloween night doing the usual candy collection from the households of their village. As the village is too big for a single group to collect the candy from all houses sequentially, Johnny and his friends have decided to split up so that each of them goes to a different house, collects the candy (or wreaks havoc if the residents don't give out candy), and returns to a meeting point arranged in advance.

There are n houses in the village, the positions of which can be identified with their Cartesian coordinates on the Euclidean plane. Johnny's gang is also made up of n people (including Johnny himself). They have decided to distribute the candy after everybody comes back with their booty. The houses might be far away, but Johnny's interest is in eating the candy as soon as possible.

Keeping in mind that, because of their response to the hospitality of some villagers, some children might be wanted by the local authorities, they have agreed to x the meeting point by the river running through the village, which is the line y = 0. Note that there may be houses on both sides of the river, and some of the houses may be houseboats (y = 0). The walking speed of every child is 1 meter per second, and they can move along any direction on the plane.

At exactly midnight, each child will knock on the door of the house he has chosen, collect the candy instantaneously, and walk back along the shortest route to the meeting point. Tell Johnny at what time he will be able to start eating the candy.

Each test case starts with a line indicating the number n of houses ($1 \le n \le 50\,000$). The next n lines describe the positions of the houses; each of these lines contains two floating point numbers x and y ($-200\,000 \le x, y \le 200\,000$), the coordinates of a house in meters. All houses are at different positions. A blank line follows each case. A line with n = 0 indicates the end of the input; do not write any output for this case.

For each test case print two numbers in a line separated by a space: the coordinate x of the meeting point on the line y = 0 that minimizes the time the last child arrives, and this time itself (measured in seconds after midnight).Your answer should be accurate to within an absolute or relative error of 10^{-5}.

Вход	**Продължение на входа**	**Изход**
2 1.5 1.5 3 0 1 0 0 4 1 4 4 4 -3 3 2 4 …	… 5 4 7 -4 0 7 -6 -2 4 8 -5 0	1.500000000 1.500000000 0.000000000 0.000000000 1.000000000 5.000000000 3.136363636 7.136363636

НБУ
НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
София 1635, ул. Монтевидео 21
тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 23 март 2013 г.

B. Working at the Restaurant

Last night, Tom went on a date with a really nice girl. However, he forgot to take his credit card with him and he had no cash in his wallet, so he ended up working at the restaurant to pay for the bill. His task is to take plates from the waiter when he comes from the tables, and pass them along when the dishwasher requests them. It is very important for the plates to be washed in the same order as they are brought from the tables, as otherwise it could take too long before a plate is washed, and leftover food might get stuck. Trying to hold all the plates in his hands is probably not a great idea, so Tom puts them on a table as soon as the waiter hands them over to him, and picks them up from the table again when the time comes to pass them along to the dishwasher. There is space for only two piles of plates on the table, which will be referred to as pile 1 and pile 2. There is only one table Tom can use.

Tom won last year's SWERC, so he is certainly capable of optimizing for efficiency. You have to output a transcript of one possible way in which Tom might decide to organize the plates on the table during the process, given the sequence of plates and requests he receives.

Input

The input has several test cases. Each case begins with a line containing a number N ($1 \le N \le 1\,000$), followed by N lines, which contain either **DROP m** or **TAKE m**, where $m > 0$ is the number of plates to take or drop. **DROP m** represents that the next event is the waiter bringing m plates to Tom, so he has to drop them on the table, while **TAKE m** represents that the next event is Tom taking m plates from the table and passing them along in the right order. You can assume that he never receives a **TAKE m** instruction when there are fewer than m plates on the table, and that the sum M of all values of m corresponding to **DROP** operations does not exceed 100 000. Note that there might be plates left on Tom's table when the last request is issued, as Tom might be relieved of his duty to stay until the restaurant closes. The input ends with a line with $N = 0$, which must not be processed.

Output

For every test case, the output will be a series of lines describing the operations to be performed with the plates. The content of each line will be one of the following:

- **DROP 1 m (DROP 2 m)**, $m > 0$, if Tom needs to take a plate from the waiter, drop it on top of pile 1 (pile 2), and repeat this operation m times in total.
- **TAKE 1 m (TAKE 2 m)**, $m > 0$, if Tom needs to take a plate from the top of pile 1 (pile 2), pass it along to the dishwasher, and repeat m times in total.
- **MOVE 1->2 m (MOVE 2->1 m**), $m > 0$, if Tom needs to take a plate from the top of pile 1 (pile 2), drop it on top of pile 2 (pile 1), and repeat m times in total.

You must output at most $6N$ lines, and the total number of movements of plates in your transcript (that is, the sum of the m's printed in your output, for all three kinds of operations), must be at most $6M$, as otherwise Tom won't be able to cope with all the work.

Note that Tom must obey the commands in the same order as they are issued. This means that, if he receives a **TAKE m** command, he must perform a certain number of **MOVE** and **TAKE** operations such that the sum of the numbers of plates **taken** adds up exactly to **m** before performing the operations corresponding to the next command; and if he receives a **DROP m** command, he must perform a number of **DROP** or **MOVE** operations for which the sum of the numbers of plates **dropped** adds up exactly to **m** before performing the operations corresponding to the next command.

Of course, it is also forbidden to take plates from the waiter or pass them along to the dishwasher in the absence of the corresponding order.

There **must** be an empty line between the outputs of different cases.

Any solution satisfying these conditions will be accepted.

Sample Input

```
3
DROP 100
TAKE 50
TAKE 20
3
DROP 3
DROP 5
TAKE 8
0
```

Sample Output

```
DROP 2 100
MOVE 2->1 100
TAKE 1 50
TAKE 1 20

DROP 2 3
DROP 2 5
MOVE 2->1 8
TAKE 1 8
```

НБУ
НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
София 1635, ул. Монтевидео 21
тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика
Школа „Състезателно програмиране”
СЪСТЕЗАНИЕ, 23 март 2013 г.

C. Happy Telephones

In the land of Eden, all phone conversations are happy ones. People complaining on the phone are immediately put in jail. To enforce this law, the police taps all phone conversations.

The police wants to hire the appropriate number of operators to listen to all conversations in a given period of time. Unfortunately, each of their operators can listen to one conversation only before needing a really long break to rest from the effort.

As a contractor of the police department, you have been asked to provide a program capable of determining the required number of operators. If the program does not work correctly, you will be put in jail as well, along with all the unhappy complainers. Do you really want to end up there?

Input

Each test case starts with two integers denoting the number of phone calls N ($1 \le N < 10\,000$) and the number of intervals M ($1 \le M < 100$). This is followed by N lines describing the telephone calls, each one consisting of four integers *Source, Destination, Start* and *Duration*. *Source* and *Destination* identify the pair of telephone numbers establishing the connection ($0 \le$ *Source, Destination* $\le 10\,000\,000$). *Start* and *Duration* are the start time and duration of the call in seconds ($1 \le$ *Duration* $\le 10\,000$ and *Start* ≥ 0). You can safely assume that the sum of *Start* and *Duration* fits into a 32-bit signed integer.

Afterwards follow M lines containing the time intervals the police are interested in, each on described by two integers *Start* and *Duration*, in the same format and with the same meaning and constraints as those in the telephone calls. The last test case is represented by $N = M = 0$ and must not be processed.

Output

For each of the M intervals of each test case, print the number of calls that are active during at least one second of the interval.

Sample Input	Sample Output
3 2 3 4 2 5 1 2 0 10 6 5 5 8 0 6 8 2 1 2 8 9 0 10 9 1 10 1 0 0	3 2 1 0

НБУ
НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
София 1635, ул. Монтевидео 21
тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика
Школа „Състезателно програмиране"
СЪСТЕЗАНИЕ, 23 март 2013 г.

Задача D. Максим

След като станал студент, Максим се преместил да живее в столицата, където се намира университетът, в който той учи. Сега той трябва да се погрижи за много нововъведения в живота си, като едно от важните е придвижването до различни части на града. А Максим обича всичко да прави по максимален начин, ето защо иска да открие максимално удобни маршрути до важни обекти, които му се налага да посещава. Той установил, че градският транспорт включва N ($1 \le N \le 10$) вида транспортни средства – автобуси, микробуси, трамваи и други. За всеки от видовете транспорт е известен броят на линиите, които той обслужва. Известен е и броят на спирките по всяка линия на градския транспорт, като общият брой на спирките е M ($100 \le M \le 1000$). След като се запознал с различни документи за организацията на градския транспорт в града, Максим успял да открие и максималното време в секунди, което е необходимо за придвижване на едно транспортно средство по една линия на градския транспорт от началната до крайната ѝ спирка. За различните линии то не е по-малко от 300 и не е по-голямо от 30000. Също така той разгледал карта на линиите от градския транспорт, от която намерил информация за броя K на спирките по всяка линия ($5 \le K \le 100$), както и номерата на самите спирки, през които преминава транспортното средство от дадена линия. Съществуват спирки, през които преминават маршрутите на различни транспортни средства. Максим проявил специален интерес към тях, тъй като от тях може да се придвижи до максимален брой различни места. Ето защо той решил, че му е необходима програма, която да познава номерата на максимално натоварените спирки, т.е. такива, през които преминават максимален брой превозни средства от градския транспорт, и да му помага да избира от тях тези, които са най-близки до дома му. Той се сетил, че сред документите за организацията на градския транспорт, видял изследване, в което се казвало, че най-натоварените спирки са с номера N и M. За щастие, тези спирки се намирали близо до дома му. Оставало само да измери времето в минути, което му е необходимо да стигне до всяка от тях.

Помогнете на Максим да определи от коя спирка може да достигне до най-голям брой обекти, които му се налага да посещава. За целта напишете програма, която по зададени N, M и времето в минути за достигане на всяка от тези две спирки, определя номера на най-подходящата за Максим спирка, като удовлетворява всички горепосочени условия.

Програмата трябва да обработва няколко тестови примера при едно извикване. На първия ред на стандартния вход е зададен броят на тестовите примери. За всеки от тях програмата чете от един ред четири цели числа – номерата на спирките M и N и времето в секунди за достигане до всяка от тях.

За всеки тестов пример програмата извежда на един ред на стандартния изход едно цяло число – номера на най-удобната, съгласно горните условия, спирка.

Вход	Изход
1 149 7 40 25	7

НБУ
НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
София 1635, ул. Монтевидео 21
тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика

Школа „Състезателно програмиране”

СЪСТЕЗАНИЕ, 23 март 2013 г.

Задача Е. К-ТОРКИ

Дадено е множество, състоящо се от n различни елемента ($0 < n < 20$). Напишете програма, която намира броя на всичките различни ненаредени k-торки ($0 < k < 51$) от негови елементи такива, че i-тият елемент на даденото множество участва във всяка k-торка точно $b_{i,1}$ пъти или точно $b_{i,2}$ пъти, ..., или точно b_{i,j_i} пъти.

Вход

Програмата чете данните от стандартния вход, като на първия ред е даден броят на тестовете. За всеки тест стойностите на n и k са написани на отделен ред, след който следват n реда. На първо място в i-тия от тези редове е записан броят на следващите числа в реда, които са стойностите на $b_{i,1}$, $b_{i,2}$, ..., b_{i,j_i} (цели числа, по-големи или равни на 0 и по-малки от 10).

Изход

За всеки тестов пример програмата трябва да изведе на отделен ред на стандартния изход търсения брой.

Вход	Изход
2 3 4 2 0 1 2 0 3 4 1 2 3 4 3 4 2 0 3 2 1 2 2 0 1	3 1

НБУ
НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
София 1635, ул. Монтевидео 21
тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика
Школа „Състезателно програмиране"
СЪСТЕЗАНИЕ, 23 март 2013 г.

Задача F. ЗАПАЗЕНА МАРКА

След мощен запой с минерална вода, Станчо и Манчо сериозно се скарали. Станчо се обзаложил на 4 каси минерална вода с Манчо, че всяко число, което завършва на 4 се дели на 2. Лукавият Манчо, обаче, посочил за контрапример числото 3,24 и отмъкнал касите. Сега Станчо е накърнен и обеднял (защото, както знаете, минералната вода поскъпна). За това той е решен да си върне водата обратно, като атакува Манчо в съда.

Манчо има много фирми, но не е регистрирал имената им като запазени марки. След анализ на действащото законодателство, Станчо установил, че може да регистрира като запазени марки низове, които се съдържат в имената на фирмите на Манчо, при което Манчо ще трябва да му изплати компенсация. За да не изглежда нагъл, Станчо смята да атакува името α на всяка фирма на опонента с точно един подниз β. По закон, при такъв иск, Манчо трябва да изплати като компенсация левовата равностойност на $n = m.l$ бутилки минерална вода от 500 ml, където m е броят на срещанията (може и застъпващи се) на β в α, а l – дължината на β.

Ако Манчо има фирма **alalabalala**, например, тогава Станчо би могъл да регистрира като запазена марка низа **a**, при което Манчо трябва да му изплати цената на 6 бутилки вода (6 срещания на низ с дължина 1 буква). Ако Станчо го атакува със запазената марка **ala**, обаче, Манчо ще трябва да му изплати цената на 12 бутилки (4 срещания на низ с дължина 3 букви). Станчо може да регистрира и цялата дума **alalabalala** (никой закон не е безгрешен), при което ще получи като компенсация цената на 11 бутилки. Напишете програма, която за всяка фирма на Манчо, пресмята цената на колко най-много бутилки вода може да получи Станчо.

Вход

На стандартния вход ще бъдат зададени имената на фирмите на Манчо. Всяко име е записано на отделен ред и представлява низ от най-много 2000 малки латински букви.

Изход

За всяко зададено на входа име на фирма, програмата трябва да изведе на на отделен ред на стандартния изход едно число – максималния брой бутилки, цената на които Станчо може да получи от атака на името на тази фирма.

Вход	**Изход**
alalabalala blablablabla stanchoeglupavood	12 18 17

НБУ
НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
София 1635, ул. Монтевидео 21
тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика

Школа „Състезателно програмиране”

СЪСТЕЗАНИЕ, 23 март 2013 г.

Задача G. КОЛОНА

При разкопки в покрайнините на село Кюлевча, изследователите от местния археологически клуб “Каквото открием за нас си е!” се натъкнали на три вида каменни блокове, с които били строени древните тракийски колони. Първия вид били с височина един метър, вторият вид – с височина два метра, а третия вид – с височина три метра. Известен факт било, че при строенето на колони, древните траки никога не поставяли два блока с еднаква височина един върху друг. Изследователите започнали да се чудят, по колко различни начина древните траки можели да построят една колона с височина N метра, използвайки трите вида каменни блока и следвайки изискването да не се поставят два блока с еднаква височина един върху друг. Друг известен факт било, че изследователите от клуба изобщо не можели да решават такива задачи и оставят това на вас.

Напишете програма, която пресмята броя на всички възможни начина за подреждания на трите вида каменни блокове един върху друг, така че да се получи колона с височина N, никои два съседни блока на която да не са от един и същ вид.

Вход

На първия ред на стандартния вход е зададено цялото число T – броят на тестовите примери. За всеки тестов пример на отделен ред е зададена височината N на колоната ($1 \leq N \leq 100$).

Изход

За всеки тестов пример, на отделен ред на стандартния изход програмата трябва да изведе броя на всички възможни подредби на трите вида каменни блока един върху друг така, че никои два съседни да не са от един и същ вид.

Пример

Вход	Изход
2 1 4	1 3

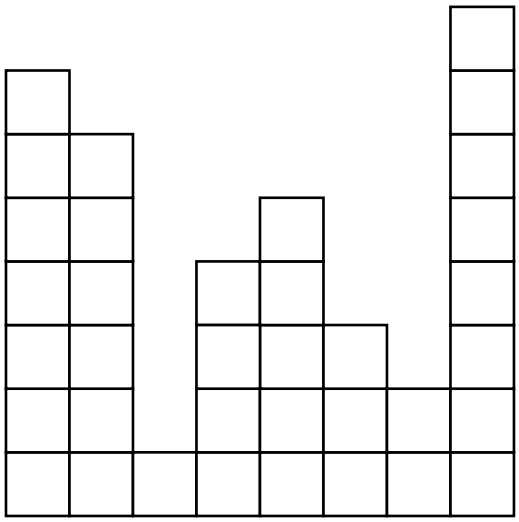
НБУ
НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
София 1635, ул. Монтевидео 21
тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика

Школа „Състезателно програмиране"

СЪСТЕЗАНИЕ, 23 март 2013 г.

Задача Н. ДЪЛГА РЕДИЦА ОТ КУЛИ



Интересът към построяване на високи сгради (кули) в познат ви морски курорт, дори при странните правила въведени от администрацията продължава да расте. Затова властите регулирали нови N еднакви парцела (сега N е значително по-голямо), разположени в редица край морския бряг и ги предложили на N строителни фирми. Всяка от фирмите бързо представила проект за построяване на съответната кула, като в проекта на всяка фирма бил заложен максималният брой етажи, които фирмата може да си позволи да построи. Оказало се, че в получената редица всичките N цели положителни числа са различни. Архитектът на курорта си имал своя идея за това как трябва да изглеждат кулите и решил да избере един интервал от редицата и да разреши строителство само на тези фирми, парцелите на които попадат в избрания интервал, като на всички фирми разреши да построят по толкова етажа, колкото е минималният брой, който някоя от фирмите в интервала е заявила. Разбира се, добре е така да се подбере интервалът, че общия брой на построените етажи да е максималният възможен. Напишете програма, която да намери този максимален общ брой етажи.

Вход

На първия ред на стандартния вход ще бъде зададен броят T на тестовите примери. На първия ред за всеки тест ще бъде зададено цялото число N ($3 \leq N \leq 1\,000\,000$). На втория ред ще бъдат зададени N положителни цели числа, не по-големи от 1 000 000 – броят на етажите които фирмите са заявили, подредени така както са подредени парцелите им на морския бряг.

Изход

На единствения ред на стандартния изход програмата трябва да изведе намерения максимален брой етажи.

Вход	**Изход**
1 8 7 6 1 4 5 3 2 8	12

НБУ
НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
София 1635, ул. Монтевидео 21
тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика

Школа „Състезателно програмиране"

СЪСТЕЗАНИЕ, 23 март 2013 г.

Задача I. ПРОИЗВЕДЕНИЕ НА ДВЕ ПРОСТИ ЧИСЛА

Днес едно от важните приложения на простите числа са шифрите за защита на информацията. При шифрите с публичен ключ, базирани на факторизацията на прости числа, публичният ключ е число M, което е произведение на две големи и различни прости числа, $M = AB$. Алгоритъмът за шифриране използва числото M, за да шифрира съобщението. Алгоритъмът за дешифриране изисква знанието на простите множители A и B. Така дешифрирането е лесно, ако са известни простите множители, които се съдържат в частния ключ и е изключително трудно, ако не са известни.

Напишете програма, която по зададено M да намира числата A и B.

Вход

На **стандартния вход**, разделени с интервал или край на ред, са зададени много цели положителни числа, всяко от които е по-малко от 2147483647.

Изход

За всяко число M от входа програмата трябва да отпечатат на отделен ред на **стандартния изход** двете намерени числа A и B, като $A < B$.

Пример

Вход	Изход
15 221 196921	3 5 13 17 191 1031

НБУ
НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
София 1635, ул. Монтевидео 21
тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика

Школа „Състезателно програмиране"

СЪСТЕЗАНИЕ, 23 март 2013 г.

Задача J. Хитрия шивач

Един скъперник искал да си ушие костюм за малко пари. Шивачът му предложил два начина на плащане:

1) да плати 217 лева за плата и ушиването на костюма;
2) да плати само за копчетата, но по следния начин: за първото копче – 1 ст., за второто 2 пъти повече отколкото за първото и т.н., за всяко следващо 2 пъти повече отколкото за предходното копче. Костюмът има 18 копчета.

Напишете програма, която по зададена цена ***k*** на първото копче и броя копчета ***n*** извежда колко трябва да плати скъперникът по втория начин в лева и стотинки.

Скъперникът се зарадвал много на втория начин на плащане и избрал него. Дали не се е излъгал?

Вход

На първия ред на **стандартния вход** е зададен броя на тестовете ***t***. Следват ***t*** отделни реда, съдържащи двете цели положителни числа – ***k*** и ***n***.

Изход

За всеки тестов пример програмата да извежда на **стандартния изход** две цели неотрицателни числа, разделени с интервал – сумата за плащане в лева и стотинки.

Ограничения: $1 \leq k < 10$, $1 < n < 61$.

Вход	Изход
2 3 4 1 8	0 45 2 55

НБУ
НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
София 1635, ул. Монтевидео 21
тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика

Школа „Състезателно програмиране”

СЪСТЕЗАНИЕ, 23 март 2013 г.

Задача К. Максимална стойност

Дадено е трицифрено число. Поставяме между първата и втората цифра, а също и между втората и третата цифра по един знак за събиране или за умножение. След това пресмятаме получения аритметичен израз.

Напише програма, която намира най-голямата стойност, която може да се получи.

Вход

Програмата трябва да може да обработва множество тестови примери, като поредният се задава на отделен ред на **стандартния вход** - едно трицифрено число.

Изход

На отделен ред на **стандартния изход** програмата трябва да изведе едно цяло число, равно на търсената най-голяма стойност за съответния тестов пример.

Пример:

Вход	**Изход**
128 103	17 4

НБУ
НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
София 1635, ул. Монтевидео 21
тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика

Школа „Състезателно програмиране”

СЪСТЕЗАНИЕ, 23 март 2013 г.

Задача L. Фигурно пързаляне

В рамките на Зимните олимпийски игри се провежда състезание по фигурно пързаляне. Състезателят се оценява като се събират оценките на журито и от получената сума се изважда най-високата и най-ниската оценка. Журито се състои от шест човека.

Напишете програма, която пресмята окончателната оценката на състезателя, ако са дадени оценките на журито. Ако журито единодушно дава еднакви оценки, да се пресметне сумата от всички оценки.

Вход

Програмата трябва да може да обработва множество тестови примери. За всеки тест от стандартния вход се въвеждат шест цели числа – оценките на всеки член на журито. Числата са разделени с по един интервал.

Изход

За всеки тестов пример на отделен ред на стандартния изход програмата трябва да изведе едно цяло число, равно на резултата на състезателя.

Ограничения

Най-високата оценка, която може да даде член на журито е 20.

Пример:

Вход	Изход
8 2 3 9 5 6	22

НБУ
НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ
София 1635, ул. Монтевидео 21
тел.: 55 81 37, 55 21 35, факс: 957 19 30

Департамент Информатика

Школа „Състезателно програмиране”

СЪСТЕЗАНИЕ, 23 март 2013 г.

Задача М. Състезания

Дядо Коледа много се интересува от състезанията по информатика. Той иска да награди най-редовните участници в състезанията. За целта всеки участник в началото на годината получава състезателен номер, валиден за всички състезания през годината. Дядо Коледа даде на джуджетата списъците с номерата на участниците във всички състезания през годината и им поръча да му направят списък с номерата на учениците участвали във всички състезания. Учениците са номерирани с числата от 1 до K ($0 < K < 221$). Помогнете на джуджетата като напишете програма, която обработва множество тестови примери, като за всеки тест въвежда информация за състезания N ($0 < N < 21$) и извежда номерата на учениците, участвали във всички състезания.

Вход

За всеки тест на първия ред на стандартния вход е зададен броя състезания N. Следват N реда - номерата на участвалите в съответното състезание ученици завършващи с 0. Края на входа е маркиран с 0.

Изход

За всеки тестов пример програмата извежда на стандартния изход в нарастващ ред номерата на учениците, участвали във всички състезания. Номерата са разделени един от друг с интервал. Ако няма такива се извежда 0.

Пример:

Вход	**Изход**
3 2 3 8 12 34 20 21 22 33 0 11 13 17 22 6 8 33 41 45 9 0 6 11 8 17 33 9 22 0 4 1 2 3 6 4 5 0 7 8 20 9 10 13 17 0 14 15 45 56 12 11 0 60 58 34 67 0 0	8 22 33 0