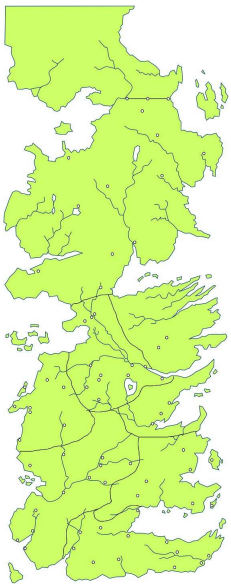
Problem A
Seven Kingdoms
Problem ID: kingdom

Jon Dayne is the ruler of a huge country called Seven Kingdoms. He has two sisters, Arya and Sansa, and wants to give some cities of Seven Kingdoms to them. He will rule the remaining cities or if no city remains, goes to the Wall, a colossal fortification along the northern border of the Seven Kingdoms, to be the Lord commander. Arya is the Lady of Winterfell and Sansa is the Lady of King's Landing. The cities in Seven Kingdoms (including Winterfell and King's Landing) are connected to each other with a network of roads (although some cities may be disconnected from the other cities, because they are either located on an island or they are currently at war with these other cities). There is no direct road between Winterfell and King's Landing and they do not share a common neighbour city.

Jon wants to assign a collection of cities to each one of his sisters such that each city in a collection is connected with a direct road to all other cities in that collection and the remaining cities, not in these two collections, are also connected with a direct road to each other. The collection assigned to Arya must include Winterfell and the collection assigned to Sansa must include King's Landing. Jon needs your help to determine whether this is possible and if this is possible, you should tell him the cities in each collection.



Input

The input consists of a single test case. The first line contains two integers n and m, where n ($2 \leq n \leq 2000$) is the number of cities, m is the number of roads. Each of the next m lines contains two integers x_i and y_i ($1 \leq x_i, y_i \leq n$) describing one road, where x_i and y_i are the distinct cities the road connects. Winterfell is city 1 and King's Landing is city 2 in the road network.

Output

If it is not possible to partition the cities in the way explained, display the word `impossible`. Otherwise, display two lines: the first containing the cities in the collection assigned to Arya and the second containing the collection of cities assigned to Sansa. If there are many such collections, any one of them is acceptable.

Sample Input	Sample Output
9 11 1 4 5 4 1 5 6 2 6 7 7 2 3 8 3 9 8 9 6 8 5 9	1 4 5 2 6 7

Sample Input	Sample Output
9 11 1 4 5 4 1 5 6 2 6 7 7 2 3 8 3 9 3 8 6 8 5 9	impossible

Every year, several universities arrange inter-university national programming contests. ACM ICPC Dhaka site regional competition is held every year in Dhaka and one or two teams are chosen for ACM ICPC World Finals.

By observing these, MMR (Mission Maker Rahman) has made a plan to open a programming school. In that school, N courses are taught. Each course is taught every day (otherwise, programmers may forget DP while learning computational geometry!). You will be given the starting time A_i and finishing time B_i (inclusive) of each course i ($1 \leq i \leq N$). You will be also given the number of students registered for each course, S_i ($1 \leq i \leq N$). You can safely assume no student has registered to two different courses. MMR wants to hire some rooms of a building, named *Sentinel Tower*, for running that school. Each room of Sentinel Tower has a capacity to hold as much as M students. The programmers (students) are very restless and a little bit filthy! As a result, when $\texttt{course}_i$ is taken in a class room, after the class is finished, it takes $\texttt{clean}_{ij}$ time to clean the room to make it tidy for starting teaching $\texttt{course}_j$ immediately just after $\texttt{course}_i$ in the same room.

Your job is to help MMR to decide the minimum number of rooms need to be hired to run the programming school.

INPUT

Input starts with an integer T ($T \leq 100$) denoting the number of test cases. Each case starts with two integers N ($1 \leq N \leq 100$), number of courses and M ($1 \leq M \leq 10000$), capacity of a room. Next N lines will contain three integers A_i, B_i ($0 \leq A_i \leq B_i \leq 10000000$) and S_i ($1 \leq S_i \leq 10000$), starting and finishing time of a course. Next N lines will contain the clean time matrix, where the i^{th} row will contain N integers $\texttt{clean}_{ij}$ ($1 \leq i \leq N$, $1 \leq j \leq N$, $0 \leq \texttt{clean}_{ij} \leq 10000000$, $\texttt{clean}_{ii} = 0$).

OUTPUT

For each case, print the test case number, starting from 1, and the answer, minimum number of rooms needed to be hired.

SAMPLE INPUT

```
3
1 5
1 60 12
0
4 1
1 100 10
50 130 3
150 200 15
80 170 7
0 2 3 4
5 0 7 8
9 10 0 12
13 14 15 0
2 1
1 10 1
12 20 1
0 2
5 0
```

SAMPLE OUTPUT

```
Case 1: 3
Case 2: 22
Case 3: 2
```

Problem C
Institute of Advanced Category Manipulation
Problem ID: acm

A certain magazine annually ranks universities across the country in various categories, as a service for readers who may be deciding which university to attend. Each university is given three scores R, T, and S (integers from 0 to 100) for its research, teaching, and community service, respectively. These scores are then interpreted somehow and the top universities in a number of categories are named. There are many categories to compete in (research university, undergraduate university, small university, etc.). Sometimes these categories appear to be invented just to make a particular university a "top university" (e.g. top university in rural area with population 30,000 to 45,000).

The Institute of Advanced Category Manipulation (ACM) is not pleased with its ranking, because it is not a top university in any category. The public relations department of ACM has decided that it can invent its own category of universities, and rank each university based on a linear combination of the three scores. The combined score U for each university is calculated by the formula:

$$U = aR + bT + cS$$

where a, b, c are constant real numbers. To make sure that the readers do not notice the scores suddenly getting too large or too small, it is required that $a^2 + b^2 + c^2 = 1$. Since some of the scores for ACM are low compared to its peers, they will even allow any of a, b, and c to be negative. A university is considered a top university in the new category if its combined score is greater than or equal to the combined scores of each of the other universities. Each category is defined by the constants (a, b, c) satisfying the constraints given above.

This idea is quickly catching on among university administrators. For example, the Research University of Ivory Towers has decided to use $(a, b, c) = (0.57735, -0.57735, -0.57735)$. Even with this manipulation, some universities cannot be considered top universities. In the second case of the sample input below, The University of Mediocrity has a score of $R = T = S = 50$. It cannot be considered a top university among the given universities regardless of how it manipulates the categories.

In this problem, you will be given the research, teaching, and community service scores of some universities. Your task is to determine which of these universities can be considered a top university.

Input

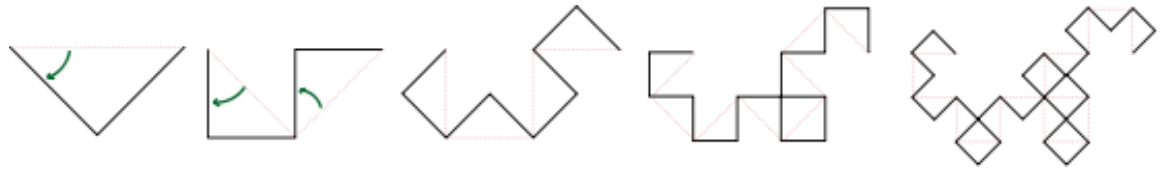
The input consists of multiple test cases. The first line of input is a single integer, not more than 10, indicating the number of test cases to follow. The first line of each case is a positive integer n ($1 \leq n \leq 50$) specifying the number of universities to be considered. The following n lines each contains three integers specifying the research, teaching, and community service scores.

Output

For each case, display on a single line a string consisting of n characters. The ith character is `T` if the ith university given in the input can be considered a top university and `F` otherwise.

Sample Input	Sample Output
2 4 100 0 0 0 100 0 0 0 100 100 100 100 9 100 0 0 0 100 0 0 0 100 100 100 0 100 0 100 0 100 100 100 100 100 0 0 0 50 50 50	TTTT TTTTTTTTF

Edward is now 21 years old. He has to appear in an exam to renew his *State Alchemist* title. This year the exam is arranged in a bit different way. There will be a long hallway. Each alchemist will enter the hallway from the left side and come out from the right side and he has to do this **n** times. During this tour they have to bend the hallway segments right-left alternatively. Let's describe the process in some pictures:



- First time (First picture): Initially, the hallway is a straight line (soft line in the first picture). So alchemist will bend this segment to right side (he is going from left to right) like the hard line in the first picture above.
- Second time (Second picture): Now he will find two segments in hallway (like soft line in picture). So he will bend the first hallway to right, second one to left (like the hard lines).
- Third time (Third picture): Now he will find four segments in the hallway (like the soft lines) and he will bend them to Right, Left, Right and Left respectively.
- And this goes on for fourth and fifth times in the picture.

Since Full Metal Alchemist Edward is so good, he did it perfectly. Now it is turn of the judges to check the bending if it is correct or not. The judge enters at the left end and comes out from the right end. But during his travel he notes down the turning, R for Right and L for Left. So if **n = 1**, then the judge would have noted down **L**. If **n = 2**, it would have been **LLR**. For **n = 4**, it would have been: **LLRLLRRLLLRRLRR**.

Since this string will grow exponentially with **n**, it will be tough to check whether the bending is correct or not. So the judges have some pre-generated strings and they know whether this string will appear as substring in the final string or not. Unfortunately the judges have lost the answer sheet, can you help them to recover it?

INPUT

First line of the test file contains a positive integer **T** denoting number of test cases (**T <= 105**). Hence follows **T** lines, each containing an integer and a string: **n S**. **n** is the number of times Edward has passed through the hallway; and **S** is the string the judge is going to check with. You may assume that **S** consists of only the letters **L** and **R**. (**n <= 1000**, length of **S <= 100**). Also you may assume that length of **S** will not be greater than the length of the string for **n**.

OUTPUT

For each test case output the case number and **Yes** or **No** denoting whether the string is in the final string as substring.

SAMPLE INPUT

```
2
1 R
4 LRRLL
```

SAMPLE OUTPUT

```
Case 1: No
Case 2: Yes
```

Problem E
Credit Card Payment
Problem ID: creditcard

Picture from Wikimedia Commons

Using credit cards for your purchases is convenient, but they have high interest rates if you do not pay your balance in full each month.

The interest rate is commonly quoted in terms of "annual percentage rate" (APR) which is then applied to the outstanding balance each month. The APR can be converted to a monthly interest rate R. At the end of each month, the monthly interest rate is applied to the outstanding balance and the interest is added to the total balance. Any payment made will be applied to the balance in the following month. The monthly interest is rounded to the nearest cent (rounding up 0.5 cent and above) in the calculations.

You have unfortunately accumulated an outstanding balance B at the end of the month and you can only afford to pay up to some amount M every month. If you do not make any more purchases with the credit card, what is the minimum number of payments needed to completely eliminate the outstanding balance? It is possible that you cannot pay off the balance in 100 years (1200 payments).

Input

The input consists of multiple test cases. The first line of input is a single integer, not more than 1000, indicating the number of test cases to follow. Each of the following lines specify the input for one case. Each line contains three positive real numbers separated by single spaces: R, B, and M. The real numbers have two digits after the decimal point, satisfying $R \leq 50.00$ and $B, M \leq 50000.00$. R is the monthly interest rate and is specified as a percentage.

Output

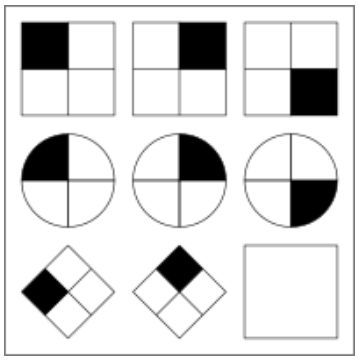
For each case, display on a line the minimum number of payments needed to eliminate the outstanding balance. If this cannot be done in at most 1200 payments, print instead `impossible`.

Sample Input	Sample Output
11 2.00 100.00 105.00 2.00 100.00 102.00 2.00 100.00 100.00 2.00 100.00 4.00 2.00 100.00 3.00 2.00 100.00 1.00 2.00 100.00 2.00 9.56 5462.50 522.22 12.50 29876.44 33610.99 5.50 1.00 1.05 14.78 40181.09 46119.86	1 1 2 36 56 impossible impossible impossible 2 2 1

Problem F
IQ Test
Problem ID: iq

In many IQ tests, the following type of questions is often given:

> Given the first few terms of an integer sequence, what is the next term?



Picture from Wikimedia Commons

For example, if you are given the sequence 1, 1, 2, 3, 5, 8, 13, 21 you may recognize this as the Fibonacci numbers and write down 34 as the next term.

There is no "correct answer" because the next term can be any integer and still be generated by a polynomial (possibly of a very high degree). In this problem, we are only interested in sequences that satisfy a recurrence relation of the form

$$f(n) = a_1 f(n-1) + \ldots a_d f(n-d),$$

where $1 \leq d \leq 3$, and $a_1, \ldots, a_d$ are integers. If the sequence satisfies multiple recurrence relations of the type above, we will always prefer one with a smaller d.

Input

The input consists of multiple test cases. The first line of input is a single integer, not more than 500, indicating the number of test cases to follow. Each case is specified on one line. Each line contains a number of integers: the number of given terms in the sequence n ($8 \leq n \leq 12$), followed by n integers containing the given sequence. Each of the given terms has absolute values at most 1000. You may also assume that the given sequence satisfies at least one recurrence relation in the form described above. The first d terms in the given sequence are non-zero, for the smallest d for which a recurrence exists.

Output

For each case, display on a line the next term generated by the recurrence relation selected by the criteria above. You may assume that the next term in the sequence has absolute value at most 100,000.

Sample Input	Sample Output
3 8 1 1 2 3 5 8 13 21 8 1 1 1 1 1 1 1 1 8 1 -2 4 -8 16 -32 64 -128	34 1 256

Vivoparc is a zoological park located in Valencia. It has recently added a new area formed by a big plane savanna grassland divided into several enclosures.

Our purpose is to assign one animal of the 4 different species (lions, leopards, tigers, and panthers) to each of the new VivoParc enclosures. These animals are very territorial therefore we must be sure that no animal can see other animal of its own species from its enclosure. The Vivoparc manager has sent us a file with the visibility from the different enclosures and we have to assign one species to each enclosure. At the end of the process, all the enclosures must have an assigned species.

INPUT

The first line of the input file contains the number of enclosures ($N \leq 100$). Each of the following lines contain a visibility restriction: 1-3 means that animals in enclosure #1 can see animals in enclosure #3 and animals in enclosure #3 can see animals in enclosure #1. Note that the Vivoparc manager is not a very well organized person and therefore, some data appearing in the file may be redundant.

OUTPUT

The output file contains one of all the possible species assignation. The file consists of a line per enclosure and each line contains the number of the enclosure followed by the assigned species (1= Lion, 2= Leopard, 3= Tiger, 4= Panther). Enclosures assignation must appear in ascending order.

SAMPLE INPUT

```
8
1-2
3-1
4-5
4-8
1-7
1-4
7-1
2-4
1-8
6-7
2-3
1-5
1-6
7-6
7-8
2-5
7-1
3-4
5-6
7-8
```

SAMPLE OUTPUT

```
1 4
2 2
3 1
4 3
5 1
6 2
7 1
8 2
```

Problem H
Chemicals Monitoring
Problem ID: chemicals

Victor works for Alberta Chemicals Monitoring (ACM). ACM is a company that analyses raw environmental data related to chemicals used in oil sand and other industries in Alberta, and produces some reports for environmental watchdogs.

Picture from Wikimedia Commons

Victor is in charge of a multi-processor cluster in ACM. Each processor is connected to a dedicated special purpose output generation unit (OGU). This cluster receives several raw data streams from field sensors and assigns each stream to a processor. Each processor performs some real time processing on a data stream and immediately after its termination, produces a report using its OGU.

Each stream has an integer starting time s, an integer duration d and a priority p. This stream is active in the interval $[s, s + d)$ (right-open interval). The report of each stream must be produced immediately after its termination; otherwise, it will be useless. An OGU creates a report extremely fast, so you can assume that an OGU produces this report instantly.

In the past, at any instance of time, the number of data streams were not more than the number of processors and OGUs. So, Victor could process all data streams. Unfortunately, recently, in a suspicious power surge, all OGUs burnt out. Victor was able to salvage one OGU by using parts from the other OGUs. Now, he can no longer produce a report for all data streams and needs to choose a subset of them based on the priorities assigned to them. To handle access to this OGU, Victor restructured the cluster architecture as follows. When a stream starts, the system either admits or rejects it. If it admits a stream, the unique identifier of the processor assigned to this stream is pushed onto the stack. Only a processor having its identifier on top of the stack can use the OGU to produce its report. After production of the report, the processor identifier is popped from the stack. It should be noted that if some streams start at the same time, he can push their processor identifier in any order of his choice. Now, Victor needs your help to choose a subset of streams such that their reports can be generated with this single OGU. The total priority of the streams in the chosen subset should be maximized.

Input

The input consists of a single test case. The first line contains an integer n, where n ($1 \leq n \leq 5000$) is the number of data streams. Each of the next n lines contains three integers s_i, d_i, p_i ($1 \leq s_i, d_i \leq 10^9$, $0 \leq p_i \leq 100,000$) describing one data stream, where s_i is its start time, d_i is the duration of the stream, and p_i is its priority. Note that the cluster has at least 5000 processors.

Output

Display the maximum total priority of a subset of streams such that their reports can be generated with the architecture described above using a single OGU.

Sample Input	Sample Output
4 1 3 6 2 5 8 3 3 5 5 3 6	13

Sample Input	Sample Output
6 5 4 10 3 4 6 1 8 100 3 2 3 4 2 4 3 2 2	115

Imagine you are in the hiring process for a company whose principal activity is the analysis of information in the Web. One of the tests consists in writing a program for maintaining up to date a set of trending topics. You will be hired depending on the efficiency of your solution.

They provide you with text from the most active blogs. The text is organised daily and you have to provide the sorted list of the N most frequent words during the last 7 days, when asked.

INPUT

Each input file contains one test case. The text corresponding to a day is delimited by tag `<text>`. Queries of top N words can appear between texts corresponding to two different days. A top N query appears as a tag like `<top 10 />`. In order to facilitate you the process of reading from input, the number always will be delimited by white spaces, as in the sample.

Notes:

- All words are composed only of lowercase letters of size at most 20.
- The maximum number of different words that can appear is 20000.
- The maximum number of words per day is 20000.
- Words of length less than four characters are considered of no interest.
- The number of days will be at most 1000.
- $1 \leq N \leq 20$

OUTPUT

The list of N most frequent words during the last 7 days must be shown given a query. Words must appear in decreasing order of frequency and in alphabetical order when equal frequency. **There must be shown all words whose counter of appearances is equal to the word at position *N*. Even if the amount of words to be shown exceeds *N*.**

SAMPLE INPUT

```
<text>
imagine you are in the hiring process of a company whose
main business is analyzing the information that appears
in the web
</text>

<text>
a simple test consists in writing a program for
maintaining up to date a set of trending topics
</text>

<text>
you will be hired depending on the efficiency of your solution
</text>

<top 5 />

<text>
they provide you with a file containing the text
corresponding to a highly active blog
</text>

<text>
the text is organized daily and you have to provide the
sorted list of the n most frequent words during last week
when asked
</text>


<text>
each input file contains one test case the text corresponding
to a day is delimited by tag text
</text>

<text>
the query of top n words can appear between texts corresponding
to two different days
</text>

<top 3 />

<text>
blah blah blah blah blah blah blah blah blah
please please please
</text>

<top 3 />
```

SAMPLE OUTPUT

```
<top 5>
analyzing 1
appears 1
business 1
company 1
consists 1
date 1
depending 1
efficiency 1
hired 1
hiring 1
imagine 1
information 1
main 1
maintaining 1
process 1
program 1
simple 1
solution 1
test 1
that 1
topics 1
trending 1
whose 1
will 1
writing 1
your 1
</top>
<top 3>
text 4
corresponding 3
file 2
provide 2
test 2
words 2
</top>
<top 3>
blah 9
text 4
corresponding 3
please 3
</top>
```

We have **N** persons and **N** vacuum cleaners. How much area can we clean? If you are thinking- "Isn't it a silly question? We can clean the entire hallway!" then you are wrong! You are forgetting that we need some electricity point in the floor to work with the vacuum cleaner. So it might not be possible to cover entire ground. Again the electrical wire of a vacuum cleaner is limited one as well. If a vacuum cleaner has wire of **D** units long, then the person using cleaner can go at most **D** unit distance away from the electrical point. But to make things worse, the cleaner have to be exactly **D** unit distance away. There is some glitch in the wire, so if the wire becomes loose at any moment (becomes less than **D** unit distance from the electrical point), the vacuum cleaner loses electricity. On top of this, the handle of the vacuum cleaner is too heavy, so the person cannot move the cleaner handle more than **d** unit distance from him. So to sum up, each cleaner has to be exactly **D** unit distance from the electric point and it can clean everything within **d** unit distance from it.

Now for each vacuum cleaner you are given **D** and **d**. Also you are given the co-ordinate of the electrical points where the vacuum cleaner is attached at. Find out the area of the ground that can be cleaned with the given setup. Please note that, some area can be covered by multiple persons but we are interested in the union of the area not sum of the area covered by individuals.

INPUT

In the first line of the input file number of test cases are given, **T** ($\leq$ **30**). Hence follow **T** test cases. Each test case starts with a positive integer **N** ($\leq$ **500**). In the next **N** lines you will be given description for the vacuum cleaners. Each line will contain, **4** integers: **x y D d**. **(x, y)** is the co-ordinate of the electrical point, **D** is the wire length and **d** is the distance of vacuum head from the person. ($|\mathbf{x}|$, $|\mathbf{y}| \leq$ **1000**, **0** $<$ **D**, **d** $\leq$ **200**).

OUTPUT

For each case print the case number and the area of the ground that can be covered. Error up to 10^{-2} will be ignored.

SAMPLE INPUT

```
2
1
0 0 10 1
2
0 0 7 1
13 0 10 1
```

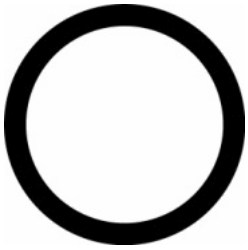
SAMPLE OUTPUT

```
Case 1: 125.663706
Case 2: 205.474931
```

EXPLANATION OF THE SAMPLES

First Case

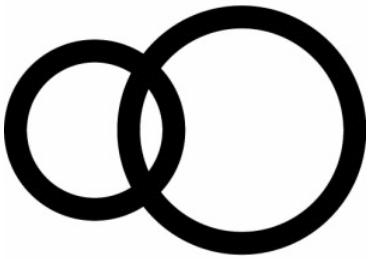
Only one cleaner. Electrical point is at **(0, 0)** and $\mathbf{D} = \mathbf{10}$, $\mathbf{d} = \mathbf{1}$. So the person handling this cleaner will be moving around a circle with **10** unit radius centering at **(0, 0)**. Since $\mathbf{d} = \mathbf{1}$, the cleaner will be able to clean any dirt with in **1** unit distance from him. The region cleaned by the cleaner is denoted by black region in the following picture:



Here the black region is bounded by two circles both centering at co-ordinate **(0, 0)**. Radii of the circles are **11** and **9**. So the area covered $= \pi \times (11^2 - 9^2) =$ **125.663706**.

Second Case

Two cleaners. Electrical point of first cleaner is at **(0, 0)** and the point for second cleaner is at **(13, 0)**. For first cleaner $\mathbf{D} = \mathbf{7}$, $\mathbf{d} = \mathbf{1}$ while for second cleaner $\mathbf{D} = \mathbf{10}$, $\mathbf{d} = \mathbf{1}$. The region cleaned by the cleaners is denoted by black region in the following picture:



Problem K
Digital Content Protection
Problem ID: content

Picture from Wikimedia Commons

Dan is working for a digital content protection company, which is responsible for the content protection of blu-ray discs based on a standard called Anti Content Misuse (ACM).

The ACM standard works as follows. Assume there are 2^n blu-ray drives/players. We represent these 2^n drives as the leaves of a complete binary tree of height n, so that each root-to-leaf path consists of n edges. Each node u in this binary tree is assigned an identifier number and contains a random key k_u. The identifier numbers are assigned as follows. The root, r, is assigned 1. In addition, the left and right children of an internal node having number i are assigned numbers $2i$ and $2i + 1$, respectively. This scheme assigns a distinct number to each node in the tree. The keys contained in the nodes are unknown to blu-ray users, but they are available to blu-ray drive manufacturers. Each blu-ray player is assigned the identifier number i ($2^n \leq i \leq 2^{n+1} - 1$) of its corresponding leaf in the tree. A manufacturer of blu-ray drives embeds the keys associated with the nodes in the path from the root to leaf number i in player number i.

To encrypt the content of a blu-ray disc, the company in charge creates a random key k called the master key. First, they encrypt k with the key k_r (recall r is the root node of binary tree) and write it on the disc as a header. Then, they encrypt the content with k, and write the encrypted data on the blu-ray disc. A blu-ray drive first decrypts the header using key k_r embedded in it and recovers the master key k and then, decrypts the content using the key k.

Unfortunately, the keys embedded in a set of blu-ray drives, R, are exposed by hackers and published on the web. As a result, we cannot encrypt the master key k using any of these exposed keys. For example, since all blu-ray drives contain k_r, the encryption scheme above does not work any more. There is a solution oversaw for this situation in the ACM standard. At the cost of a larger header, the industry can safely encrypt the content of a new blu-ray disc. They carefully choose a subset of unexposed keys K in the binary tree such that all blu-ray drives, except for drives in R, have at least one of the keys in K. They encrypt the master key k with each key $k' \in K$ and put the result in the header (i.e., there are $|K|$ ciphertexts in the header). Now, each active blu-ray drive can decrypt at least one of the ciphertexts in the header and can recover the master key k. Dan needs your help to determine a subset of keys K with minimum cardinality (which results in the smallest header) given the identifiers of hacked drives.

Input

The input consists of a single test case. A test case consists of two lines. The first line contains two integers n and $|R|$, where $1 \leq n \leq 62$ and $1 \leq |R| \leq 1000$. $|R|$ is the cardinality of R, the set of exposed drives. The second line contains $|R|$ integers, which are the identifiers of exposed blu-ray drives. You can assume that there is at least one blu-ray drive not hacked.

Output

Display the identifiers of nodes corresponding to the keys in K, satisfying the above requirements and having minimum cardinality, in increasing order and separated with single spaces.

Sample Input	Sample Output
2 1 5	3 4

Sample Input	Sample Output
3 3 10 11 12	4 7 13