# Problem A
## Thor's Successor

Time limit: 1 second
Memory limit: 256 megabytes

### Problem Description

`NCTU_Thor` is a legendary team. It is said that they chose Thor as their team name because one of the members has an extremely great thunder power, just like Thor, the god of thunder in Norse mythology. `NCTU_Thor` advanced to the 2016 ACM-ICPC (Internaional Collegiate Programming Contest) world finals last year, and finished 44th place in their first and last world finals. `NCTU_Thor` is the strongest NCTU team in the ACM-ICPC history, but they will not be able to compete for NCTU anymore. So the coach of NCTU is trying to assemble a new team which is capable to become the successor of `NCTU_Thor`.

Any team participating the ACM-ICPC must have three contestants, and the contestants must share one computer during the contest. Therefore, it seems a good idea to assemble a team which has the maximum total thunder power! Write a program to help the coach to find a threshold to eliminate all teams with insufficient thunder power.

### Input Format

The first line contains an integer $T$ ($T \leq 100$), the number of test cases. Each test case consists of two lines. The first line contains an integer $n$ ($3 \leq n \leq 60$) indicating there are $n$ members in PCCA. The second line contains $n$ integers $p_1, \ldots, p_n$ separated by blanks where $p_1, \ldots, p_n \in [1, 100]$. The $i$-th member has $p_i$ units of thunder power.

### Output Format

For each testcase, output the maximum total thunder power of Thor's successor in a line.

### Sample Input

```
3
3
1 2 8
6
1 2 3 4 5 6
9
13 22 33 54 65 76 77 78 99
```

### Sample Output

```
11
15
254
```

# Problem B
## Triangles

Time limit: 1 second
Memory limit: 256 megabytes

### Problem Description

A triangle is a shape which can be formed by connecting three non-colinear points with straight lines. Given a retangular grid formed by $n$ horizontal lines and $m$ vertical lines. How many different triangles be formed using the points on the intersections of the grid if all grid cells have the same area?

### Input Format

In first line of input, there is an integer $T$ ($T \leq 100$) indicating the number of test cases. Each of the following $n$ lines contains a test case. Each test case has two positive integers $n$ and $m$ saparated by a blank. Both of them are no more than 100, and there prodect $nm$ is no more than 1024.

### Output Format

For each test case, output an integer representing the number of such triangles.

### Sample Input

```
2
2 2
3 3
```

### Sample Output

```
4
76
```

# Problem C
## Odin

Time limit: 5 seconds
Memory limit: 256 megabytes

### Problem Description

Odin is one of the most revered god in Norse mythology. It is known that his thirst for wisdom is unending and that he would sacrifice anything for it. In order to obtain wisdom, he hanged himself, wounded himself with a spear and even stop eating and drinking for nine days and nine nights. Most famously, he took out one of his eye for the right to drink in Well of Urd, also known as the Well of Destiny. The well itself is located at the root of Yggdrasil, the tree that held the nine worlds in its branches. You see, this is how destiny works on Norse Mythology. The well is the symbol of past. and all present world grows from it. By altering the past, one can retroactively influence the present. Being able to manipulate that is the basics of magic. In theory, Odin obtained the power to shape the cosmos as he wish by completing these rituals.

If one wants to follow his path and become as powerful as he is, it is theoretically possible. One needs only to follow his path. There are $n$ rituals that must be completed and one must do them sequentially. And you may find suitable location for each ritual throughout the nine worlds. However, one must note that unlike a god, mortal lives are not particularly long. To travel through the nine worlds is not as easily as one might imagine. One cannot just take a plane from Midgard to Vanaheimr. The travel needs to be carefully planned in order to be done in one's lifetime.

### Input Format

On the first line of input there is one integer $T$ ($T \leq 70$) representing the number of test cases.

For each test case, there are three integers $n$ ($n \leq 10$), $m$ ($m \leq 300$) and $e$ ($e \leq 4000$) on the first line indicating the number of rituals that need to be performed, the number of suitable locations for them and the number of edges connecting them respectively. On the following line are $m$ integers $R_1, R_2, \ldots, R_m$ ($0 \leq R_i < n$) in which $R_i$ represents which ritual can be performed at the $i$-th location. Finally, there are $e$ lines each with three integers $u$, $v$ and $c$ ($0 \leq u, v < m$ and $c \leq 10000$). Meaning that the number of years it take to travel from location $u$ to $v$ is $c$. Note that all these edges are one directional. Since traveling between the nine worlds is not as straight forward as in Midgard (which is the world humans live in).

### Output Format

Assume paths may start at any location. Output the minimum number of years it will take to finish every ritual sequentially from number 0 to number $n - 1$ on one line. If it is impossible to finish every ritual in such order, then output `-1`.

### Sample Input

```
2
3 3 2
0 1 2
0 1 20
1 2 30
3 5 4
0 0 1 2 2
0 2 10
1 2 100
2 3 90
2 4 900
```

### Sample Output

```
50
100
```

# Problem D
## Secret Polynomial

Time limit: 2 seconds
Memory limit: 256 megabytes

### Problem Description

There is a secret polynomial $f(x)$. Its degree is less than $n$, and its coefficients are non-negative integers less than 1000000007 ($10^9+7$). That is, $f(x)$ can be written as $c_0+c_1x+\cdots+c_{n-1}x^{n-1}$ where $c_0,\ldots,c_{n-1} \in \{0,\ldots,10^9+6\}$. We will not tell you what these coefficients $c_0,\ldots,c_{n-1}$ are. Instead, we give you $n$ point-value pairs $(x_1, f(x_1) \bmod (10^9+7)),\cdots,(x_n, f(x_n) \bmod (10^9+7))$ where $x_1,\ldots,x_n$ are distinct positive integers no more than 100.

It is interesting that you can compute all coefficients from these point-value pairs. However, you are unable to determine what $c_0$ is if you only know $n-1$ point-value pairs. It's time to compute the coefficients!

### Input Format

In first line of input, there is an integer $T$ ($T \leq 20$) indicating the number of test cases. The first line of each test case contains an integer $n$ ($n \leq 50$) indicating the number of point-value pairs. The $i$-th of the following $n$ lines contains two integers $x_i$ and $r_i$ where $x_i \in [1, 100]$ and $r_i \equiv f(x) \bmod 1000000007$.

### Output Format

For each test case, output the coefficients $c_0, c_1, \ldots, c_{n-1}$ of function $f$. Separate them by blanks.

### Sample Input

```
2
2
1 2
2 3
6
1 3
2 35
3 247
4 1029
5 3131
99 509900536
```

### Sample Output

```
1 1
1 1 0 0 0 1
```

# Problem E
## Iðunn

Time limit: 5 seconds
Memory limit: 256 megabytes

### Problem Description

Iðunn is the goddess of apples of youth in Norse mythology. It is said that the gods in Norse mythology must eat apples of youth to stop aging. Loki once decoyed Iðunn out of Asgard (where the gods live). Without Iðunn's apples, the gods became old and grey. They demanded Loki to return Iðunn. Loki turned her into a nut and took her back to Asgard.

Iðunn's job is collecting apples of youth in the garden of Asgard. There are $n$ apple trees (numbered from 1 to $n$) and $m$ trails (numbered from 1 to $m$) in the garden. Each trail ends at two different apple tree. Suppose apple tree $i$ has $d_i$ trails ending at it. Iðunn will collect apples according the following procedure everyday.

1. Iðunn randomly moves to a tree with probability $\frac{d_i}{2m}$.
2. Iðunn uniformly randomly picks a trail ending at her current position, then she moves to the other end of the trail.
3. Collect apples from the tree at her position. If the tree is numbered $i$, then she should collect $a_i$ apples.
4. Repeat the previous two steps for $k$ times.

Write a program to compute the excepted number of apples daily collected by Iðunn.

### Input Format

In first line, there is an integer $T$ $(T \leq 20)$ indicating the number of test cases. For each test case, it starts with three numbers $n, m, k$ $(1 \leq n, m, k \leq 100000)$ separated by blanks in a line. There are $n$ apple trees and $m$ trails. $k$ is the parameter in Iðunn's daily collecting procedure. In the second line of each test case, there will be $n$ numbers $a_i$ $(1 \leq a_i \leq 100)$ indicating how many apples should be collected in the third step of the procedure. In the $j$-th of the following $m$ lines, there will be two numbers $u_j$ and $v_j$ $(1 \leq u_j, v_j \leq n)$ indicating that the $j$-th trail ends at tree $u_j$ and tree $v_j$.

### Output Format

For each case, output the excepted number of apples daily collected by Iðunn. An error of $5 \times 10^{-3}$ is acceptable.

### Sample Input

```
1
3 4 2
2 3 4
1 2
1 2
2 3
3 1
```

### Sample Output

```
5.75
```

# Problem F
## Pokémon GO

Time limit: 3 seconds
Memory limit: 256 megabytes

### Problem Description

Frigg is very fond of pokémon go. She wants to pick some places to go from $N$ tourist attractions and to hatch an egg during holiday. For tourist attraction $i$, Frigg has to walk $A_i$ km, and there are $B_i$ pokémons to catch. To hatch an egg in Pokémon GO, she need to walk at least $M$ km. Help her to find the most efficient way such that maximizes the ratio of the total number pokémons caught to the total distance walked.

### Input Format

An integer $T$ will be given in the first line. There are $T$ ($T \leq 10$) test cases. Each case starts with two numbers $N, M$ ($1 \leq N \leq 1000, 1 \leq M \leq 100000$). $N$ is the number of tourist attractions, and $M$ is the minimum distance to hatch an egg. In the $i$-th of the next $N$ lines, there will be two integers $A_i$ and $B_i$ ($1 \leq A_i \leq 1000, 1 \leq B_i \leq 10000$) indicating that Frigg has to walk $A_i$ km for the $i$-th tourust attraction, and there are $B_i$ pokémons to catch.

### Output Format

For each case, output the maximum ratio of the total number pokémons caught to the total distance walked. If there is no way to hatch an egg, then output `-1`. An error no more than $5 \times 10^{-4}$ is acceptable.

### Sample Input

```
1
3 10
5 7
5 3
5 1
```

### Sample Output

```
1
```

# Problem G
## Stone

Time limit: 1 second
Memory limit: 256 megabytes

### Problem Description

Alice and Bob are playing a game with some unbreakable stones. Initially there are some piles of stones, and they'll take turn splitting a pile. In each turn a player should choose a pile with more than one stones and split it into at least two equivalent piles. Alice move first, and the player who can't make a valid move lose. Who will win the game if both of them play optimally?

### Input Format

The first line contains an integer $T$ ($T \leq 20$), the number of test cases. Each test case contains two lines. The first line contains an integer $n$ ($n \leq 10000$), which is the number of piles. The second line contains $n$ integers $a_1, a_2, \ldots, a_n$ ($a_i \leq 10^9$), giving the number of stones in each pile.

### Output Format

For each test case, print "Alice" if Alice will win or "Bob" otherwise in one line.

### Sample Input

```
3
2
3 15
3
3 5 15
4
3 5 15 21
```

### Sample Output

```
Alice
Alice
Bob
```

# Problem H
## Smallest Triangle

Time limit: 1 second
Memory limit: 256 megabytes

### Problem Description

There are so many methods to calculate the area of a triangle. For example, the area of a triangle of base $b$ and height $h$ is $\frac{bh}{2}$, and it can be also represented as $\sqrt{s(s-a)(s-b)(s-c)}$ where $a, b, c$ are the lengths of its edges and $s = \frac{a+b+c}{2}$. There are also a lot of problems related to the areas of triangles. You probably have solved most of them but not this one.

This time, you have to find the smallest triangle in terms of area among a set $S$ of infinitely many triangles. Sounds hard? Just let me make the problem a little bit simpler by adding some limitation on $S$. The triangles in $S$ are exactly all the triangles which have vertices in a set $L$ of infinite points in 2D plane. Still too hard? Let me make it even simpler. The set $L$ is determined by $n$ vectors $v_1 = (x_1, y_1), \ldots, v_n = (x_n, y_n)$ where $x_1, \ldots, x_n, y_1, \ldots, y_n$ are integers. $L$ is defined as follows.

- $(0, 0) \in L$.
- If $(a, b) \in L$, then $(a + x_i, b + y_i) \in L$ for every $i \in \{1, \ldots, n\}$.

That is, any point $(x, y) \in L$ if and only if $(x, y)$ can be written as an integer linear combination of $v_1, \ldots, v_n$. More precisely, $(x, y) = (z_1x_1 + \cdots, z_nx_n, z_1y_1 + \cdots + z_ny_n)$ for some integers $z_1, \ldots, z_n$.

Now, I believe the problem is easy enough to you. Let's get it done.

### Input Format

In first line of input, there is an integer $T$ ($T \leq 500$) indicating the number of test cases. The first line of each test case contains an integer $n$ ($n \leq 100$) indicating the number of vectors. The $i$-th of the following $n$ lines contains two integers $x_i$ and $y_i$ where $x_i, y_i \in [-10^9, 10^9]$ and $v_i = (x_i, y_i)$. All points in the test case can be represented by an integer combination of $v_1, \ldots, v_n$. I.e., it can be written as $(z_1x_1 + \cdots, z_nx_n, z_1y_1 + \cdots + z_ny_n)$ for some integers $z_1, \ldots, z_n$.

### Output Format

For each test case, output the area of the smallest triangle to one decimal place. If these points cannot form any triangle, output `All points are colinear`.

### Sample Input

```
2
2
1 0
2 0
3
1 3
3 1
4 4
```

### Sample Output

```
All points are colinear
4.0
```

# Problem I
## Happy Little Tree

Time limit: 2 seconds
Memory limit: 256 megabytes

### Problem Description

Kappa has a happy little rooted tree with $n$ nodes. Kappa's friend Keepo has a favourite node on this tree. Kappa doesn't mind letting Keepo play with his tree, but he has his own rule: he only allows Keepo to traverse his tree by pre-order depth-first search. In addition, nodes are indexed by 0 to $n-1$, and each node is colored either black or white. These also limits the order of traversal. Here's the psuedocode of the traverse rule:

```
DFS(tree T):
    visit T's root u
    if(u is white):
        for(all u's child v, in ascending order by their index):
            DFS(subtree rooted at v)
    else:
        for(all u's child v, in descending order by their index):
            DFS(subtree rooted at v)
```

The root has index 0, and each non-root node has greater index than its parent. Keepo like node $n-1$ the most, but he doesn't want to spend too much time on traversing the tree. Therefore he asks you to tell him how many nodes will he visit before he reaches his favourite one. What make things even more complicated is that Kappa and Keepo change their mind frequently. Sometimes Kappa choose to change the colors of all nodes in a subtree, sometimes Keepo change his favourite node. After each change, you should also tell Keepo the new answer.

### Input Format

The first line contains an integer $T$ ($T \leq 5$), the number of test cases. The first line in each test case contains an integer $n$ ($1 \leq n \leq 100000$), the number of nodes. At the beginning, all nodes are white. The second line contains $n-1$ integers $p_1, p_2, \ldots, p_{n-1}$. $p_i$ ($0 \leq p_i < i$) is the index of node $i$'s parent. The third line contains an integer $q$ ($1 \leq q \leq 100000$), the nubmer of changes. There are $q$ lines following. In each line, there contains a character and a number $u$ ($0 \leq u < n$). If the character is 'a', it means Kappa changes the colors of all nodes in the subtree rooted at node $u$. If the character is 'e', it means Keepo changes his favourite node to node $u$.

### Output Format

For each test cases, print $q+1$ lines. The first line contains the nubmer of node visited before node $n-1$. The next $q$ lines contains the new answer after each change.

### Sample Input

```
1
4
0 0 1
2
a 0
e 2
```

### Sample Output

```
2
3
1
```

# Problem J
## Weave of Destiny

Time limit: 3 seconds
Memory limit: 256 megabytes

### Problem Description

Freyja is a goddess of love, sex, beauty, fertility and other stuff. Different from Odin, who is a Aesir god, Freyja is a member of the Vanir. These are the two opposing factions of god that had fought in the past in which the Aesir won. Freyja and her brother Freyr are the two guest hostages taken by the Aesir and lived in Asgard for a while. As an agreement between the two factions. Freyja may take half of the dead who died valiantly in battle to her afterlife realm of Folkvang, while the other half goes to Odin's hall of Valhalla. Freyja is a passionate seeker of pleasure and thrill. However, she is also a powerful practitioner of the shamanic form of magic, seidr. Seidr magic works by discerning the weave of destiny and reweaving part of it to manipulate desires, health and prosperity of others.

Although we humans are not as knowledgeable as Freyja, but practicing seidr ourselves is not entirely impossible. We know only a few things about the patterns of destiny and can only access a fairly small part of it (otherwise we will get TLE), but there is something we can do. The weave of destiny basically looks like a lot of points on a plane and they can be connected to form edges. Because we are just mere humans, we can operate only on a 2 dimensional plane. So among these points, one of the most basic way of connecting them is to make them all into triangles. Note that two edges may not cross over each other! More formally, you will be given first a convex polygon as the boundary of the scope of your manipulation. (This is a very important step in the practice of magic, to set the boundary. Disaster often happens when one neglects this basic step.) And then you will be given a set of points. With these points, you will have to cut the polygon into several triangles by connecting points to points and forming edges. However, we don't want just any triangulation. In order to produce the best result, we need a triangulation that maximizes the smallest angle in all those triangles.

### Input Format

The input will start with one integer $T$ ($T \leq 100$) on a single line representing the number of test cases.

On the next line there are two numbers $n$ and $m$. $n$ represents the number of points which made up of the boundary, and $m$ is the number of points inside it. You may assume $n+m \leq 50$ Note that the points inside the boundary may lie exactly on the boundary. The next line will contain $2n$ integers $x_1, y_1, x_2, y_2, \ldots, x_{n-1}, y_{n-1}$ separated by blanks, representing the boundary $(x_1, y_1), \ldots, (x_n, y_n)$ in counter-clockwise order. Finally there will be $2m$ integers on the last line in the same format, representing the points inside the boundary or right on the boundary. All coordinates are within the range $-100$ to 100.

### Output Format

Output the degree of the smallest angle on one line. An error less than $5 \times 10^{-3}$ is acceptable.

### Sample Input

```
2
3 0
0 0 1 0 0 1

4 2
0 0 2 0 2 2 0 2
1 1 0 1
```

### Sample Output

```
45.00
45.00
```

# Problem K
## Mímisbrunnr

Time limit: 1 second
Memory limit: 256 megabytes

### Problem Description

Mímisbrunnr is a magical well associated with Mímir who is a wisdom deity in Norse mythology. It is said that Odin drank from Mímisbrunnr and obtained wisdom and intelligence. As an ACM-ICPC contestant, you also need wisdom and intelligence to win the trophy, so you ask Mímir for drinking from Mímisbrunnr.

Mímir is wise, but he is very old. Mímir constructed the well at a perfect location, but he forgot where it is. "Odin sacrificed his right eye to exchange the right to drink. However, you are just a poor programmer, you may submit a source code to exchange my permission. If your program computes the location of Mímisbrunnr, then you may drink from it and obtain what you need to win the ACM-ICPC." Mímir tells you.

The spring from Mímisbrunnr is a mixture of water from $n$ sources located at $(x_1, y_1), \ldots, (x_n, y_n)$. Mímir built the magical well Mímisbrunnr at a location such that the total distances from there to $n$ sources is minimized. That is,

$$\sum_{i=1}^{n} \sqrt{(x - x_i)^2 + (y - y_i)^2} \leq \sum_{i=1}^{n} \sqrt{(x' - x_i)^2 + (y' - y_i)^2}$$

for any real numbers $x'$ and $y'$. If there are multiple instances achieving the minimum, Mímir would choose the one minimizing $x$. If there still are multiple instances with the same $x$, then Mímir would pick the one minimizing $y$.

It is time to obtain wisdom and intelligence equivalent to Odin's. Submit your code, then drink from Mímisbrunnr!

### Input Format

On the first line of input there is one integer $T$ ($T \leq 20$) representing the number of test cases.

For each test case, the first line contains an integer $n$ ($n \leq 10000$) indicating the number of sources of Mímisbrunnr. The $i$-th of the following $n$ lines contains two integers $x_i$ and $y_i$ which indicate the $i$-th source is located at $(x_i, y_i)$. In all test cases, you may assume that $x_i, y_i \in [-10000, 10000]$.

### Output Format

For each test case, output $x$ and $y$ in one line. Use a blank to separate them. Any error no more than $5 \times 10^{-3}$ is acceptable.

### Sample Input

```
2
2
1 0
1 1
3
0 0
1 2
2 0
```

### Sample Output

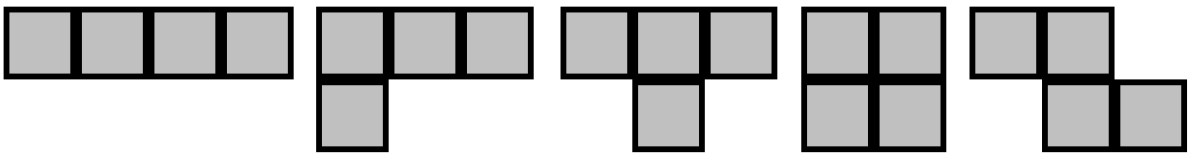
```
1.00 0.00
1.00 0.58
```

# Problem L
## Tetromino Tiles

Time limit: 1 second
Memory limit: 256 megabytes

### Problem Description

A tetromino is formed by joining four unit squared along their edges. All tetrominoes can be generated by applying rotations and reflections on the following five tetrominoes.



Skuld wants to decorate a grid wall with tetromino-shape tiles and some gems. She picks up some special grid cells, and she installs a precious gem into each of them. Skuld asks you to tile the rest of the cells on the grid wall with tetrominoes, and you have no idea how to do it. You even don't know whether it is possible to do so. It is really a challenging puzzle, isn't it?

### Input Format

The first line contains an integer $T$ ($T \leq 25$) which indicates the number of test cases. The first line of each test case contains two positive integers $R$ and $C$ which indicate the number of rows and the number of columns of the grid wall, respectively. Each of the following $R$ lines contains $C$ numbers which represent whether the corresponding cell is installed a gem or not. 1 means that the cell is installed a gem, otherwise the cell is not.

$R$ and $C$ are at most 7, and there are at most 40 grid cells which have not been installed any gem.

### Output Format

For each test case, output `Yes` if it is possible to cover the grid wall with tetromino tiles and gems. Otherwise output `no`.

### Sample Input

```
2
2 3
1 0 0
0 0 1
5 2
0 0
0 1
0 0
0 1
0 0
```

### Sample Output

```
Yes
No
```

# Problem M
## Kaprekar's constant

Time limit: 1 second
Memory limit: 256 megabytes

### Problem Description

Consider the following operation on a 4-digit number (leading zeros are allowed):

1. Rearrange the digits in descending order to get a 4-digit number $X$.
2. Rearrange the digits in ascending order to get a 4-digit number $Y$.
3. The result is $X - Y$.

The Indian mathmetician Kaprekar found that if we apply the operation repeatedly on any 4-digit number using at least 2 different digit, it will reach the fixed point 6174 in at most 7 iterations. The fixed point 6174 is known as Kaprekar's constant. Take 1234 for example. After an iteration it becomes $X - Y = 4321 - 1234 = 3087$. After the second iteration it becomes $8730 - 0378 = 8352$. After the third iteration it becomes $8532 - 2358 = 6174$. 6174 is a fixed point because $7641 - 1467 = 6174$. Note that if we choose a number whose digits are the same, the result will be 0000. For example, $3333 - 3333 = 0000$.

How about the 5-digit case? Actually some numbers don't stop at a fixed point. For instance, 87433 falls into $\{53955, 59994\}$, a cycle of length 2. Given a 5-digit number, can you figure out what will the result be after 999999 iterations?

### Input Format

The first line contains an integer $T$ ($T \leq 100000$), the number of test cases. There are $T$ lines following. Each line contains a 5-digit integer.

### Output Format

For each testcase, output the answer in a line. Leading zeros must be printed.

### Sample Input

```
3
87433
53955
33333
```

### Sample Output

```
53955
59994
00000
```