2018 National Collegiate Programming Contest
Final Round

October 6, 2018

- Problems: There are 15 tasks (36 pages in all) in this packet.
- Program Input: Input to the program are through standard input. Program input may contain one or more test cases. Test cases may be separated by any delimiter as specified in the problem statements.
- Program Output: All output should be directed to the standard output (screen output).
- Time Limit: Judges will run each submitted program with certain time limit (given in the table below).

Task Information

	Task Name	Time Limit
Problem A	Promotion Activity	1 sec.
Problem B	Approximation	1 sec.
Problem C	Struggling	1 sec.
Problem D	Unlimited 4G Mania	1 sec.
Problem E	Connecting Cities	1 sec.
Problem F	The Longest Path Tour	1 sec.
Problem G	Coloring Points	1 sec.
Problem H	Remainder 2018	1 sec.
Problem I	Data Center Locations	1 sec.
Problem J	Congruence Equations	1 sec.
Problem K	Computing Coefficients	1 sec.
Problem L	Danger Degrees	1 sec.
Problem M	Spiral	1 sec.
Problem N	Urban Planning	1 sec.
Problem O	Intelligence Organization	1 sec.

Problem A
Promotion Activity

The n stores (numbered from 1 to n) on the shopping street at an amusement park are planning a sale event. Based on prior customer survey, it is known that if store i participates in the sale event, it can expect to attract w_i shoppers to enter the store. On the other hand, if a store does not participate in the sale event, then it can expect to attract 0 shopper. However, it is not possible to have all stores to participate in the sale event. In particular, the mangement office has a rule that any two stores that participate in the sale event must be separated by at least two stores that do not participate in the sale event. In other others, if store j and store k participate in the sale event, then $|j - k| > 2$.

Given w_i of the stores, please find the maximum expected number of shoppers that can be attracted to the sale event with the above mangement office rule.

Technical Specification

1. $1 \leq n \leq 10000$
2. $1 \leq i \leq n$
3. $1 \leq w_i \leq 10000$

Input Format

The first line contains an integer, denoting the number of test cases to follow. For each case, the first line contains an integer n, denoting the number of stores. The second line contains n integers (w_1, w_2, $\ldots$, w_{n-1}, w_n) separated by a space, denoting the expected number of attracted shoppers for stores 1, $\ldots$, n, respectively.

Output Format

For each test case, output the maximum expected number of attracted shoppers for the sale event.

Sample Input

```
1
10
6 9 8 5 9 2 4 1 8 3
```

Sample Output for the Sample Input

```
26
```

Problem B
Approximation

Alice finds a simple fact that every positive integer a has a good approximate $\hat{a}_2$ so that $\hat{a}_2$ is a power of 2 and the ***ratio*** between a and $\hat{a}_2$ is at most $\sqrt{2}$ where the ratio between two positive integers x and y is defined to be

$$\max\left\{\frac{x}{y}, \frac{y}{x}\right\}.$$

Take $a = 5$ for example. If one picks 4 as the approximate $\hat{a}_2$, then the ratio between a and $\hat{a}_2$ is $1.25 \leq \sqrt{2}$. Hence, 4 is a good approximate for 5. However, if one picks 8 as the approximate $\hat{a}_2$, then the ratio between a and $\hat{a}_2$ exceeds $\sqrt{2}$, so 8 is not a good approximate for 5.

Alice continues along this direction and relaxes the requirements a little bit. She finds that every sufficiently large integer a has a good approximate $\hat{a}_{2,3}$ so that

$$\hat{a}_{2,3} \in S_{2,3} \equiv \{2^{r_2} \cdot 3^{r_3} \mid r_2, r_3 \text{ are non-negative integers}\}$$

and the ratio between a and $\hat{a}_{2,3}$ is $1 + o_a(1)$ where $o_a(1)$ denotes a function of a, say $f(a)$, so that

$$\lim_{a\to\infty} \frac{f(a)}{1} = 0.$$

Alice has no idea how fast $f(a)$ decreases to 0, and plans to have a rough estimate by plotting $f(a)$ for multiple a's. Formally,

$$f(a) = \min_{\hat{a}_{2,3}\in S_{2,3}} \max\left\{\frac{a}{\hat{a}_{2,3}}, \frac{\hat{a}_{2,3}}{a}\right\}.$$

Since the calculation is messy, could you write a program to help Alice to compute the $\hat{a}_{2,3} \in S_{2,3}$ that admits the smallest ratio for a given a?

Technical Specification

1. The given a's are integers in the range $[1, 10^9]$.

Input Format

The input is comprised of $t + 1$ lines. The first line contains an integer $t \in [0, 20]$. Each of the following t lines contains an integer $a \in [1, 10^9]$.

Output Format

For each given a in the input, output a single line containing the $\hat{a}_{2,3} \in S_{2,3}$ that admits the smallest ratio. If there are multiple choices for such an $\hat{a}_{2,3}$, output the one that has the largest value.

Sample Input

```
2
5
13
```

Sample Output for the Sample Input

```
6
12
```

Problem C
Struggling

In the sahā world (a.k.a. the world of endurance), there is full of suffering. A small positive integer called Seed wants to grow up in this world. Seed has initial value k. At each step, this pity number has two ways to change its value.

Rule A: Increase the current value of Seed by 3, which makes Seed a little larger; and
Rule B: Decrease the current value of Seed by 2 if it is greater than 2, which shrinks the value of Seed.

Seed k can choose to apply only one of the rules at each step. Unfortunately, when it becomes a multiple of 4 after the transformation, it will be immediately reduced to $k/2$ by karma (a kind of mysterious force that you cannot fight). The reduction cannot be applied again when the reduced number $k/2$ is still a multiple of 4; it is clear that karma activates at most once for each step. In addition, when Seed k was born to the world, it has at most r opportunities to apply the rules. Each application is precious because, in the end, Seed k can escape from the sahā world to the bliss pure land, a place that has only happiness, when it becomes large enough; otherwise, it has to return to the sahā world for its next life. Therefore, Seed k needs wisdom and has to choose the correct ways for applying Rules A and B at most r times in its whole life, and you know that the final result should be as large as possible.

Technical Specification

1. There are at most 10 test cases.
2. The value of Seed k is a positive integer which is less than 100.
3. The number r is a positive integer less than 10^4.

Input Format

The first line is an integer indicating the number of test cases. Each test case specifies two positive integers k and r in a line.

Output Format

For each test case, output the largest number that Seed k can become.

Sample Input

```
3
3 6
4 6
5 6
```

Sample Output for the Sample Input

```
13
17
14
```

Problem D
Unlimited 4G Mania

Dial High Telecom (DHT) announced earlier a special deal offering unlimited 4G internet with a very low price per month. As the offer is only in effect for a very short period of time, many consumers leaped into action to grab the special deal. Jenny is one of them and waiting in a long line to apply for the deal at a service station of DHT. The station has hired B clerks to meet the surge number of interested customers, and they are numbered 1 through B. Suppose the k-th clerk always takes exactly T_k minutes to serve a customer's application, and a clerk can only handle one customer at a time. Once a clerk finishes serving a client, he (she) is immediately available to serve the next customer. While the station is open, the first customer in the queue always goes to the lowest-numbered available clerk. When no clerk is available, that customer waits until at least one becomes free.

Which clerk will handle Jenny's application, if she is the N-th person in line when the station opens?

Technical Specification

1. $1 \leq B \leq 1000$
2. $1 \leq N \leq 10^9$
3. $1 \leq T_k \leq 100000$

Input Format

The first line of the input gives the number of test cases, $K = 100$. Each consists of two lines. The first contains two space-separated positive integers B and N, which are the number of clerks and Jenny's place in line, respectively. The first customer of the line is number 1, the next one is number 2, and so on. The second line contains $T_1, T_2, \ldots, T_B$.

Output Format

For each test case, output one line containing the number of the clerk who will serve Jenny.

Sample Input

```
6
2 4
10 5
3 12
7 7 7
3 8
4 2 1
3 12
4 2 1
5 1000000000
25 25 25 25 25
2 2
1 1
```

Sample Output for the Sample Input

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

Problem E
Connecting Cities

The ACM kingdom has n cities, numbered from 0 to $n-1$. An (unordered) pair $(i, j) \in \{0, 1, \ldots, n-1\}^2$ of distinct cities is said to be linkable if we are able to build a road linking i and j. It is guaranteed that if we build a road between i and j for all linkable pairs $(i, j) \in \{0, 1, \ldots, n-1\}^2$, then any two distinct cities will be reachable (via one or more roads) from each other. For every linkable pair $(i, j) \in \{0, 1, \ldots, n-1\}^2$, denote by $c_{i,j} \geq 0$ the cost of building a road between i and j. Mr. Smart wants to build roads with the minimum *product* of costs subject to every city being reachable from every other city. Please help him.

Technical Specification

1. There are at most 10 test cases.
2. $2 \leq n \leq 500$.
3. There are at most 25000 linkable pairs.
4. For every linkable pair $(i, j) \in \{0, 1, \ldots, n-1\}^2$, $c_{i,j} \in \{0, 1, \ldots, 999\}$.

Input Format

Each test case begins with n and then the number of linkable pairs. Every linkable pair $(i, j) \in \{0, 1, \ldots, n-1\}^2$ is specified by i, j and $c_{i,j}$, in that order. Furthermore, any two consecutive integers are separated by whitespace character(s). The last test case is "0 0", which shall not be processed.

Output Format

Do the following for each test case: Denoting by C the minimum product of costs subject to every city being reachable from every other city, please output the remainder of the division of C by 65537, i.e., C mod 65537.

Sample Input

```
4 5
0 2 996
1 0 3
1 2 4
2 3 800
1 3 998
3 3
0 2 500
0 1 0
```

```
1 2 600
5 7
0 1 100
0 2 999
1 2 100
1 3 10
2 3 100
3 4 58
4 1 999
0 0
```

Sample Output for the Sample Input

```
9600
0
32744
```

Problem F
The Longest Path Tour

There are 9 vertical parallel roads and 9 horizontal parallel roads in this city. Every vertical road crosses all the horizontal roads, so do the horizontal roads. Every intersection of two roads is a sightseeing spot in which a station is located. This city provides van tours to explore those sightseeing spots at a relaxing pace. Specially, the van is self-driven and is controlled by a central computer. Visitors start a tour at a station and the van drives automatically. Once a van moves near a station, the computer sends the van instructions to park. After a short stop at the station, the van starts moving toward the next station after being informed by the computer. The tour ends after the destination station is reached.

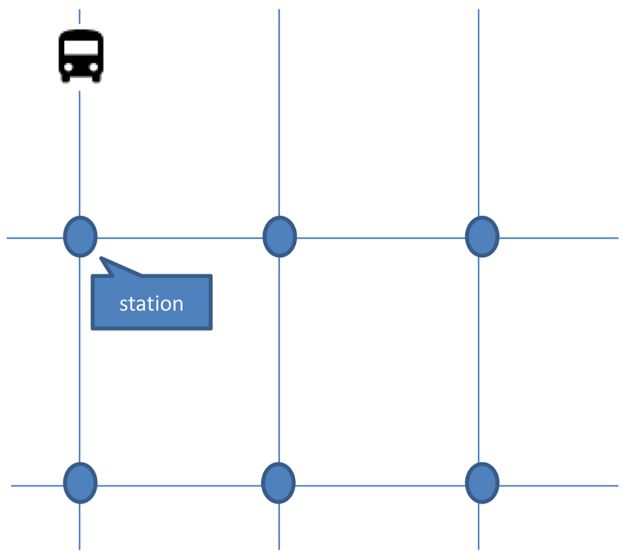


Figure 1: A part of the city.

Recently, this city offers a customized half-day tour. Visitors are asked to select two distinct stations, one to start and one to end the tour. Because a half-day tour is impossible to visit whole the city, the computer also asks the visitors to cross out 50 stations which they don't want to visit. After that, the computer arranges a tour route according to the following rules:

1. The starting station and the destination station are designated, and
2. The crossed out stations will not be visited, and
3. All roads are bidirectional, and
4. No station would be visited more than one time, and
5. The route contains as many stations as possible.

The problem is to figure out the number of stations in the tour route arranged by the central computer.

Input Format

There are 3 test cases. Each of the test case has 9 lines. Two test cases are separated by a blank line. In a test case, the n^{th} character in the m^{th} line indicates the status of the station located at the intersection of the n^{th} vertical road and the m^{th} horizontal road. A station has one of the following statuses:

1. 'S': the starting station.
2. 'D': the destination station.
3. 'X': the crossed out station.
4. '+': others.

Output Format

For each test case, output an integer in one line indicating the number of stations in the tour route arranged by the central computer. The result should include the starting station and the destination station. If there doesn't exist such tour route, output -1.

Sample Input

```
S+++XXXXX
XXX++XXXX
XXXX+XXXX
XXXX++XXX
+XXXX+++X
+++++++XX
XXX+XX+XX
XXX+XX+X+
XXX+++++D

++++XXXXX
XXXS+XXXX
XXXX+XXXX
XXXX++XXX
+XXXX+++X
+++++XXXX
XXX+XX+XX
X+++XX+X+
XXX+D++++

XXXXXS+++
XX++++XXX
XXX++XXXX
XXXX++++X
```

```
XXXXXX+++
XXXXXX+++
XXXX+++XX
XXXXXX+++
XXXX++D++
```

Sample Output for the Sample Input

```
23
-1
22
```

Problem G
Coloring Points

In Euclidean geometry, linear separability is a property of two sets of points. This is most easily visualized in two dimensions (the Euclidean plane) by thinking of one set of points as being colored black and the other set of points as being colored white. These two sets are *linearly separable* if there exists at least one line in the plane with all of the black points on one side of the line and all the white points on the other side. This idea immediately generalizes to higher-dimensional Euclidean spaces if line is replaced by hyperplane (lines in 2-d space, planes in 3-d space, etc.).

A set of points in N dimensional space is said to be in general position if no k of them lie on a $(k-2)$ dimensional hyperplane for $k \leq N+1$. For instance, in $\mathcal{R}^2$ (the Euclidean plane) you can have arbitrarily many points in general position, so long as no three of them are on the same line.

Given M points in N dimensional space (in general position), each point can be colored either white or black. Some coloring results make the points linearly separable, while some do not. For example, Fig. 1 shows four points on a plane ($M = 4$, $N = 2$). Among 16 possible ways to color the points, 14 of them are linearly separable. Please write a program to determine the number of ways to color the points so that the points are linearly separable.

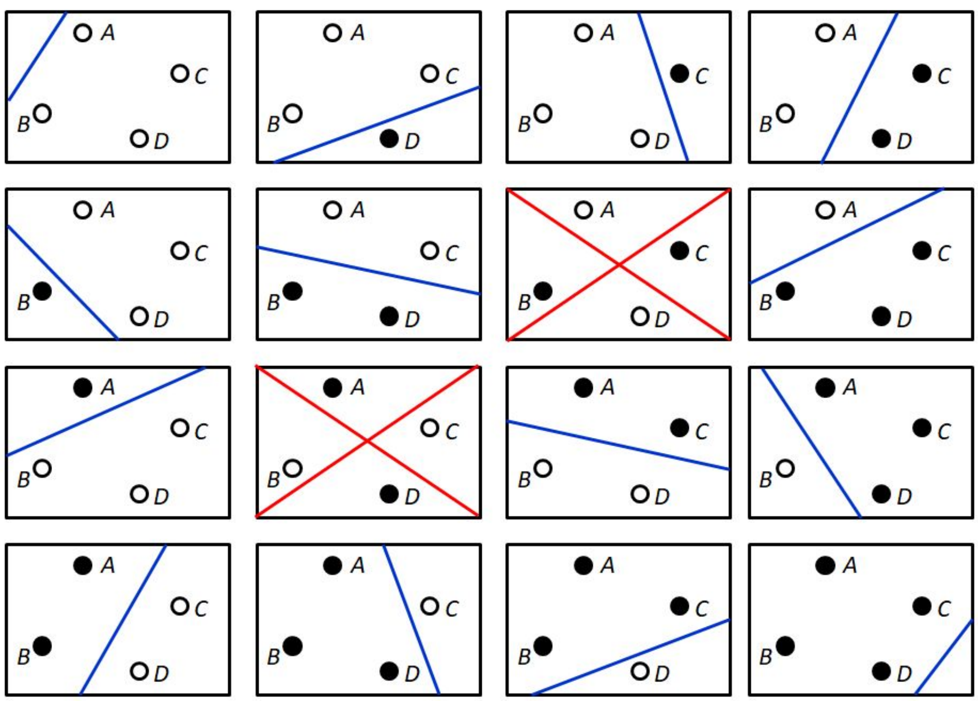


Figure 1: Illustrating linear separability

Input Format

The input contains several test cases. Each test instance consists of two positive integers, the space dimension N ($1 \leq N \leq 7$) and the number of points M ($1 \leq M \leq 60$). The zero value of N and M indicate the end of test cases and should not be processed.

Output Format

For each test case, please output an integer indicating the number of ways to color the points so that the points are linearly separable.

Sample Input

```
2 3
2 4
2 5
3 4
0 0
```

Sample Output for the Sample Input

```
8
14
22
16
```

Problem H
Remainder 2018

Given a positive integer n, n is used to build a big number concatenating all the numbers between 1 and n. If $n = 7$, then the created number would be 1234567, and if $n = 12$, then the created number would be 123456789101112. Now you are asked to calculate the remainder of the created number divided by 2018.

For example, $n = 7$, the created number would be 1234567. The remainder of 1234567 divided by 2018 would be 1569 since $1234567 = 2018 * 611 + 1569$.

Technical Specification

1. $1 \leq n < 2^{64}$.

Input Format

The first line is an integer N $(1 \leq N \leq 100)$ which indicates the number of test cases. Each test case occupies one line and contains one positive integer n.

Output Format

For each test case, please output the remainder of the created number divided by 2018.

Sample Input

```
6
7
8
9
10
11
12
```

Sample Output for the Sample Input

```
1569
1572
1603
888
19
1912
```

Problem I
Data Center Locations

Virtual Mega Service (VMS) is a well established cloud service provider. VMS plans to expand their business by building new data centers over several potential sites. After a series of thorough surveys, the CEO of VMS, Dr. Bagels, has N locations to consider. Let C_i be the cost to build a data center at location i, where $i = 1, \ldots, N$. For some pairs of locations, some profits can be generated by satisfying the demands of customers on both locations. Let P_{ij} be the profit obtained by satisfying the demand between locations i and j. Dr. Bagels wants to select some locations to build data centers such that total net profit is maximized. The total net profit is the profit obtained after deducting the cost. Your task is to write a program to help Dr. Bagels find out the maximum total net profit that can be obtained.

Technical Specification

1. N is the number of locations, $1 \leq N < 1000$.
2. M is the number of pairs of locations that produce positive profit, $1 \leq M < 10N$.
3. C_i is the cost of building a data center at location i, $1 \leq C_i \leq 10000$.
4. P_{ij} is the profit induced between locations i and j, $P_{ij} \leq 10000$.

Input Format

The first line of the input gives the number of test cases, T (< 10). For each case, the first line consists of two positive integers N and M, separated by space(s), indicating the number of locations and links that can generate profits, respectively. Then N lines follow, where the i-th ($i = 1, \ldots, N$) line has a positive integer C_i indicating the cost to build a data center at location i. Then M lines follow, where the k-th ($k = 1, \ldots, M$) line has 3 positive integer I, J, P, separated by space(s), indicating the profit generated between locations I and J is P .

Output Format

For each test case, output one line containing the maximum total net profit.

Sample Input

```
1
4 5
8
6
2
4
1 2 4
1 3 10
1 4 5
2 3 6
3 4 3
```

Sample Output for the Sample Input

```
8
```

Problem J
Congruence Equations

Many problems of science and engineering can be reduced to solving a set of simultaneous congruence equations stated as follows. Let $m_1, m_2, \cdots, m_k$ be positive integers with $m_i \geq 2$ and the greatest common divisor $gcd(m_i, m_j) = 1$ if $i \neq j$. Given integers $a_1, a_2, \cdots, a_k$, find the *minimum positive integer* x to satisfy the simultaneous congruence equations

$$\begin{aligned} x &\equiv a_1 \ (mod\ m_1) \\ x &\equiv a_2 \ (mod\ m_2) \\ &\vdots \\ x &\equiv a_k \ (mod\ m_k) \end{aligned}$$

For example, $x = 154$ $(mod\ 4 \times 5 \times 9)$ is the answer for the following congruence equations.

$$\begin{aligned} x &\equiv 2 \ (mod\ 4) \\ x &\equiv 4 \ (mod\ 5) \\ x &\equiv 1 \ (mod\ 9) \end{aligned}$$

This problem asks you to write a program to find the minimum positive integer x to solve a set of the aforementioned simultaneous congruence equations with the restrictions of $2 \leq m_i \leq 128$ for each $1 \leq i \leq k$ and $k = 3$.

Input Format

The first line contains one integer, $N \leq 8$, indicating the number of test cases to come. Each test case consists of two lines of nonnegative integers: $a_1 \ \ a_2 \ \ \cdots \ \ a_k$ and $m_1 \ \ m_2 \ \ \cdots \ \ m_k$, respectively, where $k = 3$ is fixed in this problem. The numbers in each line are separated by a space.

Output Format

For each test case, output on a line an integer corresponding to the solution of that test case.

Sample Input

```
3
```

```
2 4 1
4 5 9
1 2 4
2 5 7
0 0 1
2 5 127
```

Sample Output for the Sample Input

```
154
67
890
```

Problem K
Computing Coefficients

Let p be a polynomial with $m+n$ variables. These variables are partitioned into two parts: (1) x-variables $x_0, x_1, \ldots, x_{m-1}$, and (2) y-variables $y_0, y_1, \ldots, y_{n-1}$. Thus, the polynomial can be written as:

$$p(x_0, x_1, \ldots, x_{m-1}, y_0, y_1, \ldots, y_{n-1}).$$

In this problem, the polynomial is specified by a product of sub-polynomials. Each sub-polynomial is in the form $(u+v)$ or $(u-v)$, where u and v can be an x-variable or a y-variable. For example,

$$p(x_0, y_0) = (x_0 + y_0)(x_0 + y_0).$$

Supposed that all x-variables are fixed, then the polynomial can also be regarded as a polynomial of y-variables. In this case, the x-variable are considered as part of the coefficients. For example, let

$$p(y_0) = x_0^2 + 2x_0y_0 + y_0^2.$$

Then the coefficient of y_0 is $2x_0$, and the coefficient of y_0^2 is 1. Note that the polynomial $p(y_0)$ has 3 *monomials*, namely x_0^2, $2x_0y_0$, and y_0^2.

For another example, let $p(x_0, y_0, y_1) = (y_0 - y_1)(y_1 - x_0)(x_0 - y_0)$. After multiplying out all the sub-polynomials, the polynomial p is

$$-x_0y_1^2 - x_0^2y_0 + x_0^2y_1 - y_0^2y_1 + y_0y_1^2 + x_0y_0^2.$$

The coefficient of y_0 is $-x_0^2$, and the coefficient of $y_0^2y_1$ is -1.

In general, finding coefficient of some monomials of p may not be easy when the polynomial p is given as the product of a large set of sub-polynomials. The number of monomials grows exponentially after multiplying them out.

In this problem, assume that the polynomial is generated in a special way. Let u, v, w be 3 distinct variables chosen in the set $\{x_0, x_1, \ldots, x_{m-1}, y_0, y_1, \ldots, y_{n-1}\}$. Then they define a sub-polynomial $(u-v)(v-w)(w-u)$. All other sub-polynomials are defined in the same way. For example, the following 3 lines:

$$y_0\ y_2\ x_0$$

$$y_1\ y_0\ x_1$$

$$y_2\ y_1\ x_2$$

define a polynomial

$$(y_0 - y_2)(y_2 - x_0)(x_0 - y_0)(y_1 - y_0)(y_0 - x_1)(x_1 - y_1)(y_2 - y_1)(y_1 - x_2)(x_2 - y_2).$$

The most important thing is that, in this problem, each variable appears at most twice in all triples. Write a program to solve the problem efficiently.

Input Format

The test data contains only one polynomial but many monomials (in y-variables) whose coefficients are to be computed.

The polynomial is a product of many sub-polynomials. Each sub-polynomials is specified by 3 variables written in a line.

The first line contains 3 integers l, m, n, where l is number of lines, each line contains a triple of variables, m is the number of x-variables, and n is the number of y-variables.

The second part of the input data starts with an integer k, and the following k lines specify which monomials whose coefficients we are going to compute. Each line contains n integers, $k_0, k_1, \ldots, k_{n-1}$ which means that we are looking for the coefficient of $y_0^{k_0} y_1^{k_1} \cdots y_{n-1}^{k_{n-1}}$. For example, "2 2 2" means that we are going to compute the coefficient of $y_0^2 y_1^2 y_2^2$.

Assume that the values of k, l, m, and n are all less than 12.

Output Format

The outputs for each coefficient should be listed in many lines. Let the monomial containing $y_0^{k_0} y_1^{k_1} \ldots y_{n-1}^{k_{n-1}}$ be

$$\left(\sum_{i=1}^{t} a_i x_0^{e_{i,0}} x_1^{e_{i,1}} \ldots x_{m-1}^{e_{i,m-1}} \right) y_0^{k_0} y_1^{k_1} \ldots y_{n-1}^{k_{n-1}}.$$

For each

$$a_i x_0^{e_{i,0}} x_1^{e_{i,1}} \ldots x_{m-1}^{e_{i,m-1}}, \quad i = 1, 2, \ldots, t,$$

print t lines:

$$\begin{array}{c} a_1 \; e_{1,0} \; e_{1,1} \; \ldots \; e_{1,m-1} \\ a_2 \; e_{2,0} \; e_{2,1} \; \ldots \; e_{2,m-1} \\ \vdots \\ a_t \; e_{t,0} \; e_{t,1} \; \ldots \; e_{t,m-1} \end{array}$$

The number a_i should be printed in a field of width 4, right-justified (e. g. "`%4d`" in C `printf` format). Then followed by m $e_{i,j}$'s, $j = 0, 1, \ldots, m-1$. Print a space between each pair of adjacent $e_{i,j}$'s. In the case that $t > 0$, order them with *lexicographic order*. That is, smallest $e_{i,0}$ first, then smallest $e_{i,1}$, etc.

Finally, print a blank line after each case.

Sample Input

```
3 3 3
y0 y2 x0
y1 y0 x1
y2 y1 x2
2
2 2 2
2 1 2
```

Sample Output for the Sample Input

```
 1 0 1 2
-1 0 2 1
-1 1 0 2
 1 1 2 0
 1 2 0 1
-1 2 1 0

 1 1 1 2
-1 1 2 1
-1 2 0 2
 1 2 2 0
```

Problem L
Danger Degrees

Joe is asked to construct a new security system. During the construction, he encounters a problem as follows. Let $T = (V, E)$ be an undirected tree, in which each internal node has at most three neighbors. Each vertex $v \in V$ has a nonnegative integer weight $w(v)$ and each edge $e \in E$ has a nonnegative integer length $l(e)$. For every pair of vertices $x, y \in V$, let $d(x, y)$ denote the distance from x to y. Let $v \in V$ be a vertex. The *danger degree* of v caused by a vertex $u \in V$ is $w(u) \times d(u, v)$ and the maximum danger degree of v is $\max\{w(u) \times d(u, v) | u \in V\}$. Consider the example in Figure 1. In this example, the danger degree of vertex 3 caused by vertex 1 is $w(1) \times d(1, 3) = 1 \times 3 = 3$ and the maximum danger degree of vertex 3 is

$$\begin{aligned}
&\max\{w(1) \times d(1,3), w(2) \times d(2,3), w(3) \times d(3,3), w(4) \times d(4,3), w(5) \times d(5,3)\} \\
=&\max\{1 \times 3, 5 \times 1, 2 \times 0, 2 \times 3, 4 \times 1\} \\
=&6.
\end{aligned}$$

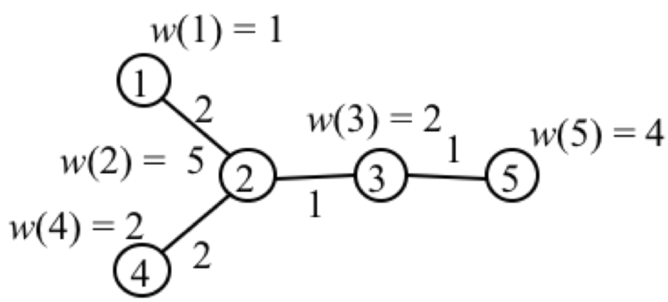


Figure 1

Let $n = |V|$ and let $m_1, m_2, ..., m_n$ be the maximum danger degrees of the vertices of T. If there are k numbers, here the median is uniquely defined to be the $\lfloor\frac{(k+1)}{2}\rfloor$-th smallest number. For example, the median of (16, 8, 6, 16, 10) is 10, and the median of (2, 10, 9, 5) is 5. In this problem, you are asked to find the median and the maximum of $m_1, m_2, ..., m_n$. For example, if the maximum danger degrees are 16, 8, 6, 16, 10, the output is 10 and 16.

Technical Specification

1. The number of vertices n is a positive integer between 2 and 40000.
2. The weight of each vertex is a positive integer between 0 and 20000.
3. The length of each edge is a positive integer between 0 and 10.
4. The maximum danger degree of a vertex is guaranteed to be not larger than a 32-bit unsigned integer capacity.

Input Format

The first line is an integer t, $1 \leq t \leq 10$, indicating the number of test cases. Each test case starts with one line containing a number n, $2 \leq n \leq 40000$, indicating the number of vertices in the tree T. Then, n lines follow, each of which contains an integer w, $0 \leq w \leq 20000$, where the ith integer is the weight of the vertex i, $1 \leq i \leq n$. And then, n–1 lines follow, each of which contains three integers i, j, l, $1 \leq i, j \leq n$ and $0 \leq l \leq 10$, indicating there is an edge of length l connecting vertices i and j.

Output Format

For each test case, output one line containing two numbers M_1 and M_2, separated by one blank, where M_1 and M_2, respectively, are the median and maximum of the n maximum danger degrees of the vertices of T.

Sample Input

```
2
5
1
5
2
2
4
1 2 2
2 4 2
2 3 1
3 5 1
4
5
3
2
4
2 3 3
2 1 7
2 4 10
```

Sample Output for the Sample Input

```
10 16
52 85
```

Problem M
Spiral

Consider a sequence of 2-dimensional points $p_1, p_2, \ldots, p_n$. Your job is to remove as few number of points as possible so that the remaining points constitute a *spiral.* That is, when you put the remaining points on the plane and connect them orderly (complied with the original sequence order) by a smooth curve, its shape looks like a spiral (see Figure 1 for an illustration). Since we are at the northern hemisphere, we are only interested in *counterclockwise spirals* in this problem. A sequence of points $q_1, \ldots, q_k$ is a *counterclockwise spiral* (or cc-spiral, for short) iff

1. The distance from q_i to the origin is strictly increasing, for i from 1 to k.

2. For any two consecutive points q_i and q_{i+1}, the vector from the origin to q_i rotates counterclockwise to the vector from the origin to q_{i+1} (excluding parallel vectors in either directions), for i from 1 to $k-1$.

3. For any three consecutive points q_i, q_{i+1} and q_{i+2}, the vector from q_i to q_{i+1} rotates counterclockwise to the vector from q_{i+1} to q_{i+2} (excluding parallel vectors in either directions), for i from 1 to $k-2$.

By default, rules 1 and 2 are true when $k = 1$ and rule 3 is true when $k \leq 2$. The input is a sequence of 2-dimensional points and the output is the length of the longest subsequence of the input that can be a cc-spiral.

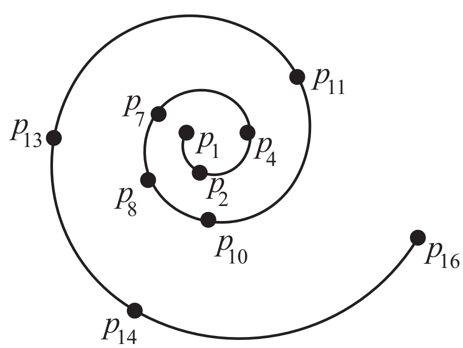


Figure 1: A spiral after removing some points.

Technical Specification

1. There are at most 10 testcases.

2. The value of each coordinate in a point is a 32-bit signed integer.

3. The number of points is at most 400.

Input Format

The first line is an integer indicating the number of test cases. For each test case, there are two lines. The first line gives you an integer n which is the number of points in the sequence. In the next line, there are $2n$ integers $x_1, y_1, \ldots, x_n, y_n$ where (x_i, y_i) specifies the ith point and two consecutive integers are separated by a space.

Output Format

For each test case, output the length of the longest subsequence of the input that can be a cc-spiral in a separate line.

Sample Input

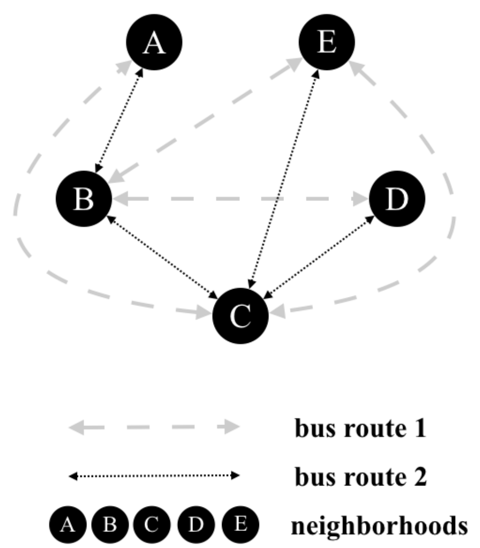
```
3
2
1 1 2 2
3
1 1 -1 1 -1 -1
5
1 1 -1 4 -2 0 0 -3 4 0
```

Sample Output for the Sample Input

```
1
1
4
```

Problem N
Urban Planning

Mayor Lee is designing the land use of his city. He divides the city into n neighborhoods, and then he finds five bus operating companies to provide public transport services. Each of the companies promises to setup a bus route to connect the n neighborhoods. To reduce the operating cost to the minimum possible while fulfilling the promise, each bus route is exactly a graph-theoretical tree that connects n neighborhoods where each edge in the tree represents a connection between two neighborhoods. Note that bus routes may have some connections in common, i.e. those connections with the same end-neighborhoods.



After setting up the bus routes, Mayor Lee plans to build a large commuting zone. A commuting zone comprises two kinds of neighborhoods, which are commercial areas and residential areas. To avoid potential traffic congestion, Mayor Lee requires each pair of commercial neighborhood and residential neighborhood in the commuting zone to have a connection by some bus route. Moreover, to secure the quality of living and working, Mayor Lee requires the number of residential neighborhoods in the commuting zone to be the same as that of commercial neighborhoods. Bus operating companies have no interest to do anything more than the promises, and therefore Mayor Lee can only build the large commuting zone based on the current bus routes. Mayor Lee wonders how large the commuting zone could be, in terms of the number of residential neighborhoods in the zone, and wonder how many choices of the largest commuting zone are. Note that two choices of the largest commuting zone may overlap because only one will be built finally. Since Mayor Lee is busy, write a program to compute his request.

In the above figure, there are six choices of the largest commuting zone, as listed in the following table.

Choice	Residential Neighborhoods	Commercial Neighborhoods
1	B, C	D, E
2	D, E	B, C
3	B, C	A, E
4	A, E	B, C
5	B, C	A, D
6	A, D	B, C

Technical Specification

- $2 \leq n \leq 1000$.

Input Format

The input comprises $6t + 1$ lines. The first line contains an integer $t \in [0, 25]$. Lines 2-7 forms the first testcase, Lines 8-13 forms the second testcase, and so on. In a testcase, the first line is an integer denoting n, and the t-th line for each $t \in [2, 6]$ contains $n - 1$ pairs of integers u_i, v_i for $i \in [1, n - 1]$ denoting that there is a connection from neighborhood u_i to neighborhood v_i in the $(t-1)$-th bus route. Note that $1 \leq u_i \neq v_i \leq n$ for each $i \in [1, n-1]$.

Output Format

Output the two requested integers "the size of the largest commuting zone" and "the number of choices of the largest commuting zone" in a line for each testcase. Since the number of choices could be large, you need to modulo the number with 2018.

Sample Input

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

Sample Output for the Sample Input

```
1 4
2 6
```

Problem O
Intelligence Organization

There is an intelligence organization that has experienced leaks and caused huge losses in money. The boss of the intelligence organization decided to reorganize the personnel in the organization. In order to increase information security, the boss intends to pick out p team leaders from all n intelligence agents (excluding the boss), and each team leader will lead at least q intelligence agents. Each intelligence agent can only contact his(her) team leader. The intelligence agent and the team leader use the text message to transmit information to each other, and each text message costs i dollar. Each team leader can contact the team members and the boss. The the team leader and the boss use the network message to transmit information to each other, and each network message costs j dollar. The process of intelligence agent A transmitting information to intelligence agent B is as follows. First, A sends a text message to the team leader of A. The team leader of A sends a network message to the boss. The boss sends a network message to the team leader of B. Finally, the team leader of B sends a text message to B. The intelligence organization is facing a financial problem. Therefore, the boss hopes that the intelligence network T^* can minimize the transmission cost while maintaining information security. In order to measure the performance of a intelligence network T, we define $d_T(u,v)$ to be the distance between u and v in T (i.e., the sum of the costs of the edges in the path between u and v in T), and define the *routing cost* of T as $C(T) = \sum_{u,v \in T} d_T(u,v)$. Given two positive integer p and q, the task of the boss is to construct a intelligence network T^* which has p team leaders and each team at least contain q team members such that the routing cost $C(T^*)$ is minimized. Your task is to write a computer program to help the boss to compute $C(T^*)$.

Technical Specification

1. $2 \leq n \leq 10000$
2. $1 \leq p \leq \frac{n}{2}$
3. $1 \leq q \leq \frac{n}{p}$
4. The cost of text message i and network message j are positive integers such that $1 \leq j \leq i \leq 10000$.

Input Format

The first line of the input contains an integer, denoting the number of test cases to follow. For each case, the first line contains five positive integers, n,p,q,i, and j, separated by a space.

Output Format

For each test case, output the corresponding cost.

Sample Input

```
1
5 2 1 2 1
```

Sample Output for the Sample Input

```
94
```