Problems and Solutions of
IOPC 2011

by

Raziman T.V. and Utkarsh Lath

Problem Code : IOPC1101

Spreadsheet scrolling[1]

Sruthi is looking at a spreadsheet containing N rows. Only K rows of the spreadsheet are visible at a time (If the top row is i, the bottom row will be i+K-1). In the beginning, rows 1...K are visible. Sruthi needs to read certain values from rows r1,r2...rM in that order. It is possible to scroll the spreadsheet so that the rows j...j+K-1 can be viewed instead of the current i...i+K-1. This operation counts as one scroll and its scroll length is defined as |j-i|

Find the minimum number of scrolls required so that Sruthi can read of all the M values in the given order. As there may be more than one way to do this, also find the minimum total scroll length required to do the reading in so many scrolls.

Input

The first line of the input contains the integer T ($\leq$20), the number of test cases to follow.

The description of each test case begins with a line containing 3 integers N($\leq 10^8$), K($\leq 10^8$) and M($\leq$50000) as defined in the problem. Following this are M lines giving the row numbers from which values have to be read sequentially.

Output

Output two space separated integers in a line per test case : The minimum number of scrolls required and the minimum scroll length required for the minimum number of scrolls.

Example

Input:

2
20 10 2
10
20
20 10 2
10
7

Output:

1 10
0 0

[1]Problem formulated by Raziman T V

Solution

A greedy approach gives the solution to the problem
At any stage, let the top row visible be i. Let the next row to be read be R_1. If $i \leq R_1 < i + K$ we do not need to scroll. So assume that R_1 is not in this range. In that case, the range of top rows that allow to view R_1 is $(R_1 - K, R_1]$. Remember this interval. Now let the next row to be viewed be R_2. Now there are two possibilities

- R_2 cannot be viewed with any row in the current interval as the top row. If this is the case, scroll to the endpoint of the remembered interval that is closer to i.
- R_2 can be viewed from a subinterval $(a, b]$ of the current interval. Update this to be the current interval and continue this process with the next row to be viewed

Problem Code : IOPC1102

Chocolate distribution[2]

In Dystopia, chocolates are being distributed to children waiting in a queue. The distribution proceeds as follows. Each chocolate bar is rectangular in shape with integer edge lengths. If the chocolate bar is a square, it is given away completely to the first child in the queue. Otherwise the largest possible square piece of chocolate is broken off from the chocolate bar and given to the first child. After a child receives his share of chocolate, he leaves the queue. The remaining portion of the chocolate bar is dealt with in the same fashion and the whole or a portion of it is given to the next child in the queue.

For example, if we start with a 5×3 chocolate bar, the first child in the queue receives a 3×3 chocolate bar, leaving a 2×3 bar. The second child gets a 2×2 bar while the third and fourth children get 1×1 bars. Thus four children have been fed using the 5×3 bar.

The Dystopian government has got a carton of chocolate bars to be distributed to children in the country. To make sure that maximum inequality is achieved while distributing chocolates, the chocolate bars in the carton are all of different sizes. For every i such that $M\leq i\leq N$ and every j such that $P\leq j\leq Q$ (where M,N,P,Q are integers) there is exactly one chocolate bar of length i and breadth j in the carton. Here a bar of length i and breadth j is considered to be different from a bar of length j and breadth i.

Given the values of M,N,P,Q find the number of children that can be fed with the chocolate in the carton.

Input

The first line of the input contains the number of test cases, T ($\leq$1000)

Following this are T lines, each describing a test case with four integers M,N,P,Q separated by spaces ($1\leq M\leq N\leq 10^8$, $1\leq P\leq Q\leq 1000$)

Output

Output T lines, each containing an integer : The number of children that can be fed using the chocolate in the carton.

Example

Input:

2
1 2 1 2
3 4 4 5

Output:

6
14

[2] Problem formulated by Raziman T V

Solution

Given the dimensions of a bar, (say a × b), the number of pieces that will be cut out can be computed easily using the formula:

$$f(a,b) = f(b, a\%b) + \lfloor \frac{a}{b} \rfloor$$

The point to note is that P and Q are pretty small, so the solution can be written as

$$\sum_{i=P}^{Q} F(i,N) - F(i,M-1)$$

where

$$F(x,y) = \sum_{j=0}^{y} f(x,j)$$

If we can find $F(x,y)$ efficiently, then we are done. So precomute and store the value of $F(x,y)$ for $x, y \leq 1000$. For larger numbers,

$$F(x,N) = F(x,x) + \sum_{y=x+1}^{N} f(x\%y, x) + \lfloor \frac{y}{x} \rfloor$$

which can be written as

$$F(x,x)t + F(x, N\%x) + \frac{xt(t-1)}{2} + t(N\%x)$$

where $t = \lfloor \frac{N}{x} \rfloor$. The complexity is $O(P-Q)$ per test case.

Problem Code : IOPC1103

Truly magical numbers[3]

Prashant thinks that the number 142857 is magical. When he multiplies the number with 2, he gets 285714 which is a permutation of the digits of the original number. On multiplying with 3,4,5 and 6 too he observes a similar behaviour.
Now Prashant wants to find out truly magical numbers. A truly magical number of order N is a number P with the following properties:

- P should have N digits and the first digit should be non-zero
- On multiplying the number with any i such that $1 \le i \le k$, the result should be a permutation of the digits of P
- There should not exist any N digit number Q (without leading zeros) with the property that for all i such that $1 \le i \le k+1$, $Q \times i$ is a permutation of digits of Q (Here k has the same value as in the previous point)
- P should not have consecutive zero digits anywhere

It is easily seen that P=142857 is indeed a truly magical number of order 6. P has no leading zeros or consecutive zeros appearing anywhere. For $1 \le i \le 6$, $P \times i$ is a permutation of digits of P. Also, there does not exist any number Q such that for all $1 \le i \le 7$, $Q \times i$ is a permutation of digits of Q.
Given an integer N, find a truly magical number of order N. Note that for certain values of N there may be more than one truly magical number of order N. In such a case, it is enough to output any one of the truly magical numbers of order N.

Input

The first line of input contains an integer T (≤ 100), the number of test cases to follow.
Following this are T lines, each containing an integer N, $250 \le N \le 40000$

Output

Output T lines, each containing a truly magical number of order N

Example

The input limits are such that an example cannot be shown here. However, to explain the formatting of the input and output, we show here a sample case that has N outside the input limit range.

Input:

```
2
2
6
```

Output:

```
13
142857
```

[3]Problem formulated by Raziman T V

Solution

The first observation we make is that K can never be greater than 9. We now show that for large numbers, including all numbers in the input range, K is equal to 9.
Notice that 142857 on multiplication with 7 gives 999999. That is, 142857 are the recurring digits of the decimal expansion of $\frac{1}{7}$. If the decimal expansion of $\frac{a}{b}$ contains a minimal length recurring block P of $b-1$ digits, then $2P$, $3P$, ...$9P$ will be permutations of the digits of P unless $9P$ has more digits than P itself. We also need $10a > b$ so that P has no leading zeros. This gives us two conditions:

- $10a > b$
- $9a < b$

Together, there give the first values of (a, b) as $(2, 19)$ and $(3, 29)$. The values for P in these cases are 105263157894736842 and 1034482758620689655172413793. Appending these numbers one after the other, we can make larger numbers with the same property. These numbers are 18 and 28 digits long respectively. So appending them multiple times, for every even number $N > 208$, a truly magical number of order N can be made. To get an truly magical number of odd order, append a zero to the end of a truly magical number with order $N - 1$.

Problem Code : IOPC1104

Triangle equality[4]

Consider three distinct points A,B,C on a plane. The sum of straight line distances from A to B and B to C is always greater than or equal to the straight line distance from A to C. Equality holds only when ABC is a degenerate triangle. This is the famous **triangle inequality**.
In this case, distance between points is measured by the Euclidean metric. ie, the distance between points (x_1, y_1) and (x_2, y_2) is given by $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. However, this is not the only metric possible. Another common metric used is the **Manhattan metric** where the distance between the pair of points is given by $|x_1 - x_2| + |y_1 - y_2|$.
You are given N distinct points on a plane where distances are measured using the Manhattan metric. Find the number of ordered triplets of distinct points (A, B, C) such that the sum of distances from A to B and B to C is equal to the distance from A to C.

Input

The first line of input contains an integer T ($\leq$10), the number of test cases to follow.
Following this are the descriptions of T test cases. Each test case description begins with an integer N ($\leq$50000), the number of points. Following this are N lines, each giving the x and y coordinates of a point ($0 \leq x_i, y_i \leq 108$) separated by a space.

Output

Output T lines, each containing the number of ordered triplets of distinct points in every test case with the given property

Example

Input:

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

Output:

2
0

[4]Problem formulated by Raziman T V

Solution

Consider B, the second point in the triplet. We have 4 possibilities for (A, C) for the given B.

- A is to the top left of B while C is to the bottom right of B
- A is to the top right of B while C is to the bottom left of B
- A is to the bottom left of B while C is to the top right of B
- A is to the bottom right of B while C is to the top left of B

Here notice that the last two sets are same as the first two on interchanging A and C.
If we have the counts of points in each quadrant relative to each point, the problem can hence be solved.
To find the counts of points to the bottom left of all points in $O(N \log N)$ time, sort all points by (x, y) lexicographically. Now sweep through the set of points in this order storing the number of points below each y coordinate in a segment tree or a trie.
However, if triplets are calculated by this method, we will end up double counting those triplets where all points share the same x or y coordinate. These have to be corrected for.

Problem Code : IOPC1105

Military patrol[5]

Dystopia consists of N cities. There are one-way roads connecting some pairs of cities. The dysfunctional state has recently seen a lot of protests to overthrow the tyrannical ruler and the government plans to use military patrol vehicles to make sure that the protests are suppressed. Every patrol vehicle is assigned a sequence of cities. If a patrol vehicle is assigned the cities c_1, c_2,...c_k then it starts from the city c_1 and takes the direct one-way road to c_2, from there it takes the one-way road to c_3 and so on. Finally the vehicle takes the one way road from c_k to c_1. This routine is repeated everyday to keep the protestors perpetually under fear.
Now note that:

- Every city has to appear in exactly one vehicle's patrol sequence exactly once
- Every patrol vehicle has to move - so it has to be assigned more than one city

The government does not have any limit on the number of patrol vehicles it can use. However, they want to make sure that the least possible amount of money is spent on the patrol mission and hence they want to minimise the total distance travelled by the patrol vehicles.
Given the road network of Dystopia, find the minimum total distance the patrol vehicles need to move so that all the cities can be patrolled. If it is impossible to organise a nationwide patrol with the given constraints, report the same.

Input

First line of the contains T, the number of test cases (T≤10)
This is followed by the descriptions of the T testcases. The first line of the description contains two integers N and R, the number of cities and one way roads respectively (N≤200,R≤10000). The cities are numbered 1,2,3,...N This is followed by R lines, each representing a one way road by 3 integers N_1, N_2 and D : the start city, the end city and the length of the road respectively ($N_1 \neq N_2$, $1 \leq D \leq 10^6$). You are assured that there is no more than one one way road from any N_1 to N_2

Output

For each test case output one line. If the patrol can be done, output the minimum total distance that the patrol vehicles have to travel. Otherwise output Impossible

Example

Input:	**Output:**
2 3 3 1 2 1 2 3 1 1 3 1 4 6 1 2 2 2 3 2 3 4 2 4 1 2 1 4 1 3 2 1	Impossible 6

[5] Problem formulated by Raziman T V

Solution

This can be modelled as min cost maximum flow problem. Build a bipartite graph G' where every vertex of the original graph $G = (V, E)$ has two copies, on the left and one on the right. For an edge (u, v) in the original graph, add an edge from left copy of u to right copy of v in G' of the same cost. G can be decomposed into cycles iff the bipartite graph G' has a perfect matching. This is because for a given decomposition of G into cycles($G'' \subset G$), there exists a matching of size $|V|$, by matching every vertex v on left to the right copy of the only vertex adjacent to v in G''. Similarly, given a perfect matching, one can decompose the graph into cycles. Also the minimum cost matching will be the minimum cost decomposition of the graph into cycles. Once the bipartite graph has been generated, finding the minimum cost perfect matching can be done using the Hungarian algorithm.

Problem Code : IOPC1106

Partitioning the plane[6]

You are given the coordinates of 4K+5 points on a plane such that no three of them are collinear. You need to select five points from these : a central point O and four arm points A,B,C,D such that:

- Rays from the centre to the arm points divide the plane into four regions containing an equal number of points
- None of the four central angles is a reflex angle
- Sum of absolute values of the cotangents of the central angles is as low as possible

If it is possible to choose points satisfying this condition, output the lowest possible value for the sum of absolute values of the cotangents of the central angles. Otherwise report that it is not possible.

Input

The first line of input contains an integer T ($\leq$4), the number of test cases. Following this are the descriptions of the T test cases.
The first line in the description of each test case gives K($\leq$100). Following this are 4K+5 lines giving the x and y coordinates of each point separated by a space ($0 \leq x,y \leq 10^6$)

Output

For each test case output in a different line the minimum sum of absolute values of the cotangents of the central angles, with six digits after the decimal point. If the division cannot be done in the manner explained, print Impossible

Example

Input:

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

Output:

```
4.500000
Impossible
```

[6]Problem formulated by Raziman T V

Solution

Assume that one of the points has been fixed as the centure O. Now sort the remaining points according to the angle it forms around the centure. Do a radial sweep choosing the i^{th}, $K+1+i^{th}$, $2K+2+i^{th}$ and $3K+3+i^{th}$ points as A,B,C,D and find the best combination. For fixed O, this can be done in $O(n \log n)$ time. Repeat the procedure taking every point as the possible centure. Overall, this gives the complexity $O(n^2 \log n)$.

Problem Code : IOPC1107

Leaky containers[7]

The acid manufacturing company has a special room to store leaky acid containers. The container holders in the room, which have the capacity to hold one leaky container each, are arranged in a rectangular grid of R rows and C columns such that the columns are in the North-South direction while the rows are in the East-West direction. Currently there are N leaky containers in some of the holders and M more have just arrived and need to be placed in the holders.

The company has realised that the containers being produced these days are exceptionally leaky. So much so that the acid that is leaking is corroding the holders completely.

Every acid container leaks either in the North-South direction or the East-West direction. Containers can be rotated by 90 degrees and thus a container that is leaking in the East-West direction can be made to leak North-South and vice versa. Given enough time, a leaky container can corrode the holder completely and start corroding the two adjacent holders in the leak direction and this process can go on.

The company employee has to make a decision fast. He needs to rotate some of the existing containers and place the new containers in proper holders and directions such that the total number of holders that will be corroded is minimised.

Find out the minimum number of holders that will be corroded after proper placement of the new containers and proper orientation of all containers.

Input

The first line of input contains an integer T ($\leq$4), the number of test cases. For each test case, the first line has four numbers R, C, N and M ($1 \leq R,C \leq 100$, $1 \leq M,N \leq 20$, $M+N \leq R*C$). This is followed by N lines, each giving the location and leak direction of an existing container by 3 integers r (row number), c (column number) and d (1 if leakage is N-S, 0 if E-W). Numbering of rows and columns begins with 1.

Output

For each test case, output on a different line the smallest number of holders that will get corroded after rotating the existing containers and placing the newly arrived containers.

Example

Input:	**Output:**
2 4 6 4 4 1 2 0 2 4 0 3 2 1 3 5 1 50 50 5 10 1 35 1 17 44 0 17 46 1 42 35 1 42 46 0	12 148

[7]Problem formulated by Prof. Manindra Agarwal

Solution

Use brute force over all 2^N possible ways of arranging the containers. Once the orientation of N existing containers are decided, some cells get corroded. Place the newly arrived containers in the corroded cells first. After that one is again left with an $R' \times C'$ matrix where he has to place containers at any place to minimize the number of cells corroded. Since R,C$\leq$ 100, and $M \leq 20$, the solution for all such cases can be precomputed using a DP. Repeat this for all possible orientations of existing containers and find the best one.

Problem Code : IOPC1108

Progressive progressions[8]

An arithmetic progression is a sequence of numbers a_1,a_2,...a_n such that $a_{i+1} - a_i$ is equal for all 0≤i<n. This difference is called the common difference of the arithmetic progression.
Now consider a sequence of arithmetic progressions $A_1 = (a_{1,1}, a_{1,2}, ...a_{1,n_1})$,$A_2 = (a_{2,1}, a_{2,2}, ...a_{2,n_2})$,...$A_k = (a_{k,1}, a_{k,2}, ...a_{k,n_k})$
A progressive progression is such a sequence with the additional properties that:

- $a_{i,n_i} = a_{i+1,1}$ for $1 \leq i < k$
- c_i, the common difference of A_i, is a positive factor of a_i,1 for $1 \leq i \leq k$
- $c_i < c_{i+1}$for $1 \leq i < k$
- $n_i > 1$ for $1 \leq i \leq k$
- $k \geq 1$

Find the number of progressive progressions such that $a_{1,1} = 1$ and $a_{k,n_k} = N$. As this number can be quite large, output it modulo 100000007.

Input

The first line of input contains an integer T (≤100), the number of test cases. This is followed by the description of the testcases. The description of each testcase consists of a single integer N ($1 <$ N $\leq 10^6$).

Output

For each testcase, output modulo 100000007 the number of progressive progressions such that $a_{1,1} = 1$ and $a_{k,n_k} = N$

Example

Input:

2
5
10

Output:

1
6

[8]Problem formulated by Raziman T V

Solution

Let $F(n, k)$ denote the number of Progressive progressions ending at n such that all common differences are less than equal to k. Then $F(n, k) = \sum_{i|n, i \leq k} \sum_{j < n/i} F(i * j, i)$. This can be computed iteratively for every k from 1 to n in $O(n/k)$ time.

Problem Code : IOPC1109

Move the books[9]

Sheldon and Lenard are a pair of nerds playing an unimaginatively named game, "Move the books". The game board is an infinitely long strip of cells numbered 1,2,3.... from left to right. On certain cells, their favourite physics books have been placed. A player's move consists of taking any one of the books and moving it to any cell which lies to its left. But there is a constraint that you are not allowed to make your book jump over a cell that contains a book already (ie, You cannot move a book from cell j to cell $i < j$ if there is a cell k which contains one or more books such that $i < k < j$). However, you can place a book into a cell even if it contains one or more books already. But books that are placed in a cell are stacked in the order in which they arrive and hence only the topmost book (the last arrived one) can be moved from there. The players make moves alternately, and the person unable to move any book loses.
They have been playing the game for a long time. Sheldon makes the first move in all the games and wins most of the time. Lenard is fed up and wants to make the first move. However, Sheldon doesn't yield and this leads to an argument. This is the final agreement they have come up with:

- They start with N books placed in different cells. The arrangement is computer generated and hence there is no player role in this step
- Lenard picks two natural numbers a & c while Sheldon picks a natural number b. Both are unaware of the number(s) the other person has chosen while choosing their own number(s). Three more books are now added to the set : a cells to the right of the rightmost current book, b cells to the right of this book and c cells to the right of the latter book.
- They start the game with the same rules as earlier, with Sheldon making the first move.

Now Lenard feels that there might be certain pairs (a,c) such that independent of which number Sheldon chooses, Lenard is assured to win the game. Given the initial configuration of the board find all such pairs, sort them lexicographically [(a1,c1) < (a2,c2) iff a1 < a2 or (a1=a2 and c1 < c2)] and output the K^{th} such pair. If there are less than K pairs with this property, output Impossible

Input

The first line of input contains an integer T (≤ 50), the number of test cases. Following this are the descriptions of the T test cases
The first line in the description of each test case contains two space separated integers N (≤ 1000) and K ($\leq 10^8$). Following these are N lines, each containing the location of a book. The book positions are given in increasing order and will each fit in a 32 bit signed integer.

Output

For each test case output the K^{th} lexicographically smallest pair of integers that will assure Lenard a win. The two integers should be separated by a space and pairs for each test case should be output on a new line. If for any test case there are less than K pairs of integers that assure Lenard a win, on the line for that test case output Impossible

Example

Input:	**Output:**
1 1 1 1	1 1

[9]Problem formulated by Raziman T V

Solution

This is just a Nim game in disguise. The value of Grundy number for a game where books are kept at d_0,d_1,...d_n will be $(d_n - d_{n-1}) \wedge (d_{n-2} - d_{n-3}) \wedge ...(d_{n-2i} - d_{n-2i-1})...$. Here $\wedge$ denotes the xor operation. So, for given values of a, b, and c, let G denote $a \wedge c \wedge (d_{n-1} - d_{n-2}) \wedge (d_{n-2} - d_{n-3}) \wedge ...(d_{n-2i+1} - d_{n-2i})...$ which is the Grundy number. Clearly, for any value of a, there is a unique value of c for which G is zero. Also there will be exactly one value of a when c becomes 0. This will be $a = (d_{n-1} - d_{n-2}) \wedge (d_{n-2} - d_{n-3}) \wedge ...(d_{n-2i+1} - d_{n-2i})...(= R)$. So for $K \geq R$, output $K, R \wedge K$ and for $K > R$, output $K + 1, R \wedge (K + 1)$.

Problem Code : IOPC1110

Road trip[10]

Phileas Fogg and Passepartout are now going on a road trip in their brand new car. They start at location A_0 and need to go to A_N. Their car has a capacity to hold only C units of fuel and can travel unit distance on unit amount of fuel. They start by filling some amount of fuel from the filling station at A_0. On the way, there are several filling stations $A_1, A_2, ...A_N$. The cost of fuel is not the same at all filling stations. Find the minimum amount that they have to spend on fuel to make the journey. Note that it is assured that the journey can be completed with the car of the given capacity.

Input

The first line of input contains an integer T ($\leq$10), the number of test cases. Following this are the descriptions of the T test cases

The first line in the description of each test case contains two space integers N ($\leq$50000) and C ($\leq 10^8$). This is followed by N lines, each containing an integer. The integer on the ith line is the distance from A_0 to A_i and is $\leq 10^8$. The distances are in increasing order. This is followed by N more lines, each containing an integer. The integer on the ith line is the cost of one unit of fuel at the filling station A_{i-1} and is $\leq 10^8$.

Output

Output one integer per test case, the minimum total amount that needs to be spent on fuel to complete the journey

Example

Input:

```
1
5 15
10
20
30
40
50
1
2
1
2
1
```

Output:

```
60
```

[10]Problem formulated by Utkarsh Lath

Solution

Consider the problem of going from city A to city B. Consider the city C which (strictly) lies between A and B and has minimum cost.
If cost at C is smaller than cost at A, then you would like to reach C with your fuel tank empty, then soln(A,B)=soln(A,C)+soln(C,B).
If distnce from A to C is greater than K, then still you will like to reach C with empty fuel tank, so it is again soln(A,B)=soln(A,C)+soln(C,B).
If distance from A to C is smaller than K and petrol is cheapest at A, then fill you tank full (or dist(B)-dist(A), whichever is lesser) and reach C with some free petrol.
But we don't know about the solution when there is some free petrol as well. So let's try again with this additional parameter: Find The minimum cost to travel from A to B when you have some free fuel at A.
If cost at C is smaller than cost at A, then you will like to reach C with fuel = max(free-(dist(C)-dist(A)),0), so soln(A,b,free)=soln(A,C,free) + soln(C,B,max(free-(dist(C)-dist(A)),0))
If distance from A to B is more than K, then you will reach C with empty fuel. so soln(A,B,free) = soln(A,C,free) + soln(C,B,0)
If none of the two cases appear, then soln(A,B,free)=soln(C,B,max(free-(dist(C)-dist(A)),0)) + min(dist(C) - dist(A)-free,0)*cost of petrol at A.
So the only tricky part now left is to find C efficietly. This can be done using range minima query. So the total Time would be $O(n \log n)$.

Problem Code : IOPC1111

Giant fountain[11]

The Dystopian government has installed a giant fountain in front of the parliament building. The fountain consists of N levels stacked one on top of the other and is situated on top of a large tank of infinite capacity. The levels of the fountain are numbered 1 to N from top to bottom. The top l_1 levels are identical with capacity c_1, the next l_2 levels identical with capacity c_2,... the final l_K levels with capacity c_K. Here $l_1 + l_2 + ...l_K = N$.

When water is added to level i beyond its capacity, the excess water overflows to level $i + 1$. Water overflowing from level N is collected in the tank. Water is added to the levels in the following fashion. First, w_1 amount of water is added to each level i such that $s_1 \leq$ i $\leq e_1$. Then w_2 amount of water is added to each $s_2 \leq$ i $\leq e_2$... Finally w_m amount is added to $s_M \leq$ i $\leq s_M$. Note that water might be added to the same level multiple times in this fashion. You have to find out the amount of water that has overflowed to the tank at the bottom, and the total number of levels of the fountain that are completely filled

Input

The first line of input contains an integer T ($\leq$10), the number of test cases. Following this are the descriptions of the T test cases
The first line of the description of a test case contains space separated integers N ($\leq 2 \times 10^8$), K ($\leq$2000) and M ($\leq 10^4$). Following this are K lines, each containing a space separated pair of integers. These are the (l_i, c_i) pairs as explained in the problem statement. Here $l_1 + l_2 + ...l_K = N$ and $c_i \leq 10^8$. This is followed by M lines, each containing a space separated triplet of integers. These are (s_i, e_i, w_i) as explained in the problem statement. $1 \leq s_i \leq e_i \leq$ N and $w_i \leq 10^6$

Output

For each test case output a space separated pair of integers : The total amount of water that has overflowed to the tank and the number of levels of the fountain that are completely filled.

Example

Input:

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

Output:

```
5 5
```

[11] Problem formulated by Raziman T V

Solution

Realize that the input $< s_i, e_i, w_i >$ can be broken into disjoint intervals s'_j, e'_j, w'_j . Also, a series of levels with same capacity can be considered as an interval in which w_i is negative. Next, one can traverse all intervals from top to bottom to find the filled levels and total overflow.

Problem Code : IOPC1112

Sister cities[12]

Unlike Dystopia, the neighbouring nation of Utopia believes in economic development. To improve the economy of the nation, the Utopian government has decided to select some pairs of cities as sister cities and take steps to improve trade relations between each pair.
There are N cities in Utopia, numbered 1 to N. There are two-way roads connecting some pairs of cities. The total number of roads in Utopia is R. Now the road network in Utopia has been created efficiently so that there is no road that is redundant. That is, there is exactly one way to travel between any pair of cities without using the same road twice. Now when a pair of cities is chosen as sister cities, the government wants to make sure that there is a direct road between them. Also, a given city cannot have more than one sister. Given the road network of Utopia, find the number of ways of selecting K pairs of sister cities under these constraints. As the answer can be quite large, output it modulo 100000007.
For example, assume that there are 6 cities in Utopia. There are direct roads between the following pairs of cities : (1,2), (2,3), (2,4), (4,5), (4,6). Notice that there is exactly one way to travel between any pair of cities. If the government wants to select two pairs of sister cities, it can do it in four ways : {(1,2),(4,5)}, {(3,2),(4,5)}, {(1,2),(4,6)}, {(3,2),(4,6)}

Input

The first line of input contains an integer T ($\leq$10), the number of test cases. Following this are the descriptions of the T test cases.
The description of each test case begins with a line containing 3 space separated integers N ($<$400), R ($<$10000) and K ($<$400). Following these are R lines, each representing a road in Utopia. The line will contain two different space separated integers N1 and N2 implying that there is a two way road between N1 and N2. You are assured that the road network has the property as described in the problem statement.

Output

For each test case, output modulo 100000007 the number of ways of selecting K pairs of sister cities satisfying the conditions in the problem statement.

Example

Input:

```
2
3 2 1
1 2
2 3
6 5 2
1 2
2 3
2 4
4 5
4 6
```

Output:

```
2
4
```

[12]Problem formulated by Raziman T V

Solution

The problem is to find number of matchings of size K in a tree. Let the tree be rooted at some vertex v_0. For a node v of tree, let $f(v,k)$ denote the number of matchings of size k in the segment of tree rooted at v, and let $g(v,k)$ denote the number of matchings in this subtree where vertex v is free. Let the children of v in the tree be $u_0, u_1..u_r$, then

$$g(v,k) = \sum_{a_0+a_1..a_r=k} f(u_0,a_0)f(u_1,a_1)...f(u_r,a_r)$$

and

$$f(v,k) = g(v,k) + \sum_{a_0+a_1..a_r=k-1} \sum_{i=0}^{r} g(u_i,a_i) \prod_{j\in[0,r],j\neq i} f(u_j,a_j)$$

Problem Code : IOPC1113

Ski slopes[13]

A skier wants to ski down from the top of a mountain to its base. There are several possible routes, using different slopes enroute, and passing through some flat areas. The effort expended in skiing down a slope depends upon the length of the slope and the speed of skiing. For each slope, there is a maximum advisable speed. The skier wants to use a toute that minimizrs the average effort spent per unit distance traveled (i.e. the total effort expended divided by the total distance traveled).
The flat regions on the mountain are numbered 1 to N from top to bottom. The skier begins at level 1 and needs to reach level N. You are given the numbers of the flat regions each slope connects. Note that on a slope, one can only ski downwards. For each slope, you are also given the length of the slope and the maximum advisable speed for it. The effort expended in skiing down a particular slope is given by the following formula:
e = d*(70-s) if s $\leq$ 60, and e = d*(s-50) if s $>$ 60
where e is the effort required, d is the distance traveled and s is the speed of travel.
You have to determine the minimum average effort per unit distance that the skier has to expend in order to reach the mountain base, while staying within the maximum advisable speed at every slope.

Input

The first line of input contains an integer T ($\leq$20), the number of test cases. Following this are the descriptions of the T test cases.
For each test case, the first line of input gives the number of flats, N (N $\leq$ 1000), and the number of slopes, R (R $\leq$ 20000), connecting them respectively. Each of the next R lines describes a slope by giving: the numbers of the flats at the top and the bottom of the slope, the maximum advisable speed for the slope ($\leq$ 100), and the length of the slope ($\leq$ 1000) respectively.

Output

For each test case, output a single number (with four places after the decimal point, rounded up) that gives the minimum average effort per unit distance that needs to be expended to ski down from the mountain top to the base. The output for each test case should be on a separate line.

Example

Input:

1
4 5
1 4 30 60
1 2 50 40
1 3 60 20
2 4 60 50
3 4 50 50

Output:

14.4445

[13]Problem formulated by Prof. Manindra Agarwal

Solution

In this problem, the optimal substructure property does not hold. That is, let there be a route from x to y. The path which optimizes average effort to reach y and passes through x need not be the path which minimizes the average effort to reach x. For example, if there is a path to x of average effort 10 and total length 100 and another path of average effort 5 and total length 10. and then length of $x-y$ route is 100 and it's average effort is 20, then it makes sense to choose the first path to x and then go to y. However, if you are given a value of average effort and asked whether it is possible to attain atmost this average effort, then it can be easily tested. Let k be the average value that you want to test. Then let $f(z)$ denote the smallest value of $\sum_{(x,y) on P} dist(x,y) * (effort(x,y) - k)$ over all paths P from which start at the top and end at x. The value of $f(z)$ can be computed for all z using the relation

$$f(z) = min_{(y,z) \in R} f(y) + dist(y,z) * (effort(y,z) - k)$$

If the f value for base is negative, then k is attainable, otherwise it is not. One can do binary search to find the minimal value fo k for which a solution is possible.

Problem Code : IOPC1114

Place-name game[14]

Place-name game is a favourite pastime among the few children that go to school in Dystopia. The game is played as follows : One player says the name of a city and the next player has to say the name of a city that begins with the last letter of the said city. The game then goes on.
Dystopian cities recently went through a massive renaming. Now each city has a name that begins with a consonant and ends with a consonant.
Anaximander is a student with a very poor knowledge of geography. Hence he fares very poorly in the game. He has recently come up with a new idea. He would just remember the name of 21 Dystopian cities. He wants to choose the 21 cities such that there is exactly one city name starting with each consonant and exactly one city name ending with each. This would give him a good advantage in the game, whether he is playing first or second.
Given the names of the N cities in Dystopia, find out the number of ways Anaximander can select 21 city names out of the lot satisfying the properties. As this number can be very large, output it modulo 100000007.

Input

The first line of the input contains N (≤1000), the number of cities. This is followed by N lines, each containing the name of a city in Dystopia. Each city name will begin and end with a consonant, and will contain at least 2 and at most 10 letters.

Output

Output modulo 100000007 the number of ways Anaximander can choose 21 city names out of the N with the intended properties.

Example

Input:

23
BBBB
CCCC
DDDD
FFFF
GGGG
HHHH
JJJJ
KKKK
LLLL
MMMM
NNNN
PPPP
QQQQ
RRRR
SSSS
TTTT
VVVV
WWWW
XXXX
YYYY
ZZZZ

[14] Problem formulated by Piyush Srivastava

BAAC
CAAB

Output:

2

Solution

Form a matrix of 21 * 21 size, where (i,j) th element contains the number of cities whose name starts with i^{th} consonant and ends with j^{th} consonant. To choose 21 distinct cities would mean that one has to choose 21 distinct rows such that one cell is chosen from every row and every column. The number of ways of choosing these 21 cities is the permanent of this matrix. The permanent can be computed in $O(n * 2^n)$ time. http://en.wikipedia.org/wiki/Computing_the_permanent

Problem Code : IOPC1115

Enumeration of rationals[15]

It is well known that rational numbers form a countable set. Hence the set of rational numbers in the open interval (0,1) also form a countable set.
Here we enumerate the rationals in (0,1) in the following fashion. First, every rational is expressed in the lowest terms : ie, as p/q where p and q are positive integers with no common factor other than one. Then we sort the fractions in the ascending order of p+q. In case of a tie, the smaller fraction comes first.
The first few terms in this enumeration are 1/2, 1/3, 1/4, 2/3, 1/5, 1/6, 2/5...
Given a natural number N, find the numerator and denominator of the Nth term in the enumeration.

Input

The first line of the input contains T ($\leq$1000), the number of test cases. This is followed by T lines, each containing an integer N($\leq 10^{11}$).

Output

For each value of N, output separated by space the numerator and denominator (in lowest terms) of the Nth fraction in the enumeration.

Example

Input:

2
3
6

Output:

1 4
1 6

[15]Problem formulated by Utkarsh Lath and Raziman T V

Solution

First realize that for a given value of $p+q$, there are $\lfloor \phi(p+q)/2 \rfloor$ distinct values of the pair (p,q). For the given limits, the sum $p+q$ will be less than 10^6, so compute $\sum_{i=1}^{n} \phi(i)/2$ for $n \le 10^6$. To find the n^{th} rational number $\frac{p}{q}$, find the sum $p+q$ using a binary search on the precomputed array of $\sum \phi(i)$. Next, we have to find the j^{th} number coprime to $p+q(=S)$, which will be the value of p. To do this, do a binary search again. Now given a number i and the S, we want to find the number of integers which are coprime to S and are less than i. let prime factors of s be $p_1, p_2..p_k$. Using inclusion exclusion, the number of such coprime integers will be $i - \sum_{p_i} \frac{i}{p_i} + \sum_{p_i,p_j} \frac{i}{p_i p_j} - ...(-1)^k \frac{i}{p_1 p_2 ... p_k}$. Computing this sum takes $O(2^k)$ time. However, realize that $k =$ the number of distinct prime factors of S, can be atmost 7 for $S \le 10^6$. So the overall complexity will be $O(\log n + 2^k \log n)$, which fits for the given time limit.

Problem Code : IOPC1116

Counting the teams[16]

The teacher in the Dystopian School for Politics and other Dirty Games (DSPDG) is training students in group activities. She feels that to really understand group behavior, students need to practice working in groups of different sizes. This is how she groups the students:
There are N students in the class, with roll numbers from 1 to N($2 \leq N \leq 10^{12}$). The teacher generates using her laptop a random permutation of the roll numbers. The student with the roll number equal to the ith number in the permutation is assigned as a "target" to the student with roll number as i (Note that "targetship" is not mutual. If 1 is the target for 2, 2 need not be the target for 1). If any student is assigned himself as the target, the teacher generates another permutation till no student is assigned himself.
The N students stand far from each other. Now student 1 goes and joins his target. After this, student 2 (and any student who is with him) joins 2's target. At the ith step, student i and anyone who is already with him joins i's target. In case i's target is already with him, nothing is done.
By following this procedure, when all students have joined their targets, the class gets split into some groups. For example, assume that there are 6 students in the class and the permutation that has been generated is {2,4,5,6,3,1}. First, 1 goes and joins 2. Then 1 and 2 join 4. Then 3 joins 5. Then 1, 2 and 4 join 6. 5 is already with 3 and hence does not move. Similarly 6 is already with 1 and does not move. In the end, we have 2 teams : {1,2,4,6} and {3,5}
Given N, find out the expectation value of the number of teams that will be formed when the teacher groups the class in this fashion.

Input

First line of the input contains T(≤ 100), the number of test cases. Following this are T lines, each containing an integer N($2 \leq N \leq 10^{12}$).

Output

For each N, output the expectation value of the number of groups formed. Output 6 digits after the decimal point while printing the expectation value.

Example

Input:

2
3
4

Output:

1.000000
1.333333

[16]Problem formulated by Utkarsh Lath and Raziman T V

Solution

Compute the expected number of cycles in a random derangement. Using linearity of expectation, the solution is

$$\frac{\sum_{i=2}^{N} \binom{N}{i}(i-1)!D(N-i)}{D(N)}$$

where $D(N)$ is number of derangements on n objects. Justification: Let there be a random variable for every possible cycle which is 1 if the cycle occurs in the derangement and 0 otherwise. The expected number of cycles is sum of the expectation value of all the cycles. There are $\binom{N}{i}(i-1)!$ cycles of length length i, and each such cyle has expectation value of $D(N-i)/D(N)$. Now rewrite this sum as

$$\frac{\sum_{i=2}^{N} \frac{1}{i} \frac{D(N-i)}{(N-i)!}}{D(N)/N!}$$

After some mathematics, this can be written as

$$\frac{\sum_{i=0}^{n-2} \frac{(-1)^i}{i!} (H(n-i)-1)}{D(N)/N!}$$

Where $H(n) = \sum_{i=1}^{n} \frac{1}{i}$ Now realize that for large values of n, the denomiator is *very very* close to $\frac{1}{e}$. And in the numerator, only first few terms would make any significant contribution. Therefore, for small n, evaluate in $O(n)$ time. For large N, find the sum of first 100 terms of numerator and multiply by e. The value of $H(n)$ can be found using some standard approximation like ramanujan's approximation for $H(n)$.