

Task: ASC

Ascending Paths



Maratona de Programação, Day 1. Available memory: 256 MB.

26.01.2015

You are visiting the Royal Botanical Gardens. In the gardens there are n intersections connected by m roads. The intersections are numbered from 1 to n . Each road connects two intersections and has an *attractiveness level*, which denotes how beautiful the plants along it are considered to be. The attractiveness level is a positive integer; the greater the better.

You want to find a walk in the gardens which will be as long as possible. You may start and finish at any intersections, and you may visit an intersection more than once during the walk. However, the walk must be *ascending*: that is, the attractiveness level of each visited road must be greater than that of the previously visited road.

Input

In the first line of the input there are two integers n and m ($1 \leq n \leq 200\,000$, $0 \leq m \leq 200\,000$), specifying the number of intersections and roads in the gardens. The next m lines describe roads; the i -th of these lines consists of three integers a_i , b_i and c_i ($1 \leq a_i, b_i \leq n$, $1 \leq c_i \leq 10^9$), specifying that there is a bidirectional road between intersections a_i and b_i with an attractiveness level of c_i . There can be multiple roads connecting the same pair of intersections, or a road with both ends entering the same intersection. It is possible for roads to go through tunnels or over bridges (that is, the graph of intersections and roads doesn't have to be planar).

Output

In the only line of the output should be one integer standing for the maximal number of roads that can be visited during an ascending walk in the gardens.

Example

For the input data:

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

the correct result is:

```
4
```

Explanation of the example: The longest ascending walk visits intersections 4, 2, 3, 4 and 5 (in that order). The attractiveness levels of the visited roads are 2, 3, 5 and 6, respectively.

Task: DNA

DNA



Maratona de Programação, Day 1. Available memory: 256 MB.

26.01.2015

The Limonian scientists specialize in genetic engineering that they carry out at local flora. Lately, they have tried to cross-breed orange and lime to get the perfect fruit for juice production. To achieve this, the scientist induce different mutations in fruit DNA chains and create new DNA chains. However, not all of them turn out to be stable. Your task is help in marking the stability of DNA chains.

In DNA chains there are purines (adenine and guanine) and pyrimidines (cytosine and thymine). According to Limonian scientists, pyrimidines are susceptible to changes only if there are many purines in their neighbourhood. For every connected fragment of the DNA chain we define its *volatility measure*. If such a fragment does not include any pyrimidines, it is fully stable, which means its volatility measure is 0. Otherwise, the volatility measure of the DNA chain fragment is defined as ratio of the number of purines in it (denoted shortly by letters **A** and **G**) to the number of pyrimidines in it (denoted by letters **C** and **T**). The volatility measure of the DNA chain is the highest possible volatility measure of any of its connected fragments.

The scientists want you to tell them how many purines and how many pyrimidines there are in the most volatile DNA chain fragment. If there are many of such fragments, scientists are interested in the longest one.

Input

In the first line of the input there is one integer t ($1 \leq t \leq 10$) specifying the number of test cases. In every of t further lines there is n -letter word ($1 \leq n \leq 1\,000\,000$) consisting of letters **A**, **C**, **G** and **T**. The word stands for the DNA chain given to you by Limonian scientists for analysis

Output

The output should consist of t lines containing answers to consecutive input test cases. The answers to the test case is a pair of numbers meaning, respectively, the number of purines and the number of pyrimidines in the longest of the most volatile DNA chain fragments.

Example

For the input data:

```
2
ATAG
TATA
```

the correct result is:

```
3 1
2 1
```

Explanation of the example: The most volatile fragment in the first DNA chain is the whole chain (with volatility measure of 3), and in the second one it is the ATA fragment (volatility measure of 2).

Task: LIS

Postmen



Maratona de Programação, Day 1. Available memory: 256 MB.

26.01.2015

Recently a new post office was built in Bytevillage. Two postmen were hired, and every day they take off from the post office building to deliver letters to the people in the village. You have to plan their routes in such a way, that the latest letter will be delivered as fast as possible.

In Bytevillage there are n houses numbered with integers from 1 to n . The post office building is a house with number 1. Houses are connected with a connected network consisting of $n - 1$ bidirectional roads. It takes 1 minute for a postman to travel along one road.

Input

In the first line of the input there is an integer n specifying the number of houses in the village ($1 \leq n \leq 3000$). In the next $n - 1$ lines there are descriptions of roads. Each description consists of two integers a, b specifying a road connecting houses with numbers a and b ($1 \leq a, b \leq n$).

Output

In the only line of the output you have to write the minimal time in minutes needed by two postmen to deliver all letters.

Example

For the input data:

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

the correct result is:

```
4
```

Explanation of the example: One postman can deliver letters to houses 2, 3 and 4 (it will take 3 minutes) and the other one to houses 6 and 5 (it will take 4 minutes).

Task: MAX

Maximum Slice



Maratona de Programação, Day 1. Available memory: 256 MB.

26.01.2015

You are given an array A of n integers A_0, A_1, \dots, A_{n-1} . A pair of integers (p, q) , such that $0 \leq p \leq q < n$, is called a *slice* of array A . The *sum* of a slice (p, q) is the total of $A_p + A_{p+1} + \dots + A_q$. The *maximum sum* of array A is the maximum sum of any slice of A .

For example, for array $A = (3, 2, -6, 3, 1)$ a slice $(0, 1)$ has sum 5. This is the maximum sum of A .

You can perform a single swap operation on array A . This operation takes two indices i and j , such that $0 \leq i \leq j < n$, and exchanges the values of A_i and A_j . The goal is to find the maximum sum you can achieve after performing a single swap.

For example, after swapping elements 2 and 4, you will get the array $A = (3, 2, 1, 3, -6)$. After that, $(0, 3)$ is a slice of A that has the sum 9. This is the maximum possible sum of A after a single swap.

Write a program which returns the maximum possible sum of A after a single swap operation.

Input

In the first line of the input there is one integer n ($1 \leq n \leq 100\,000$) specifying the length of the array. In the second line there is a sequence of n integers A_0, A_1, \dots, A_{n-1} that specifies the values in the array ($-10\,000 \leq A_i \leq 10\,000$).

Output

The output should have one line containing one integer which stands for the maximum possible sum of array A after a single swap operation.

Example

For the input data:

```
5
3 2 -6 3 1
```

the correct result is:

```
9
```

Task: PIZ

Pizzerias



Maratona de Programação, Day 1. Available memory: 256 MB.

26.01.2015

Limonia is a city full of programmers and everyone knows that pizza is the main food of programmers. Therefore, it is no surprise that a new chain of pizzerias wants to enter the Limonian market. Your task is to develop sales plan that will maximize the chain's profit.

The Limonian communication network is not really complicated. There are n junctions in the city that are numbered from 1 to n . At every junction there is one building, where programmers work, craving for pizza. Some of the junctions are connected with two-way sections of roads. It is possible to drive from one junction to any other (though, perhaps, driving through other junctions) despite the fact, that there are only $n - 1$ road sections.

Because of the antitrust regulations, the Mayor of Limonia permitted letting commercial spaces for pizzerias in no more than k buildings. Moreover, in every building only one commercial space can be let. The rent costs are known for every building.

To reach as many customers as possible, you can hire one delivery man in each pizzeria. Every delivery man should be given the route he should take to deliver pizzas to other buildings. The delivery man takes off from the junction, next to which is the pizzeria building where he is hired, and after delivery, he must return to this junction. Every two consecutive junctions in his route must be connected with a road section. Delivery man can visit particular junction more than once. He can as well stay in pizzeria and not take off at all (it means that the route consisting of one junction is also correct). Of course the delivery man must be paid too — for every road section we know the cost of driving it.

With the help of sophisticated statistical methods the revenues from sales of pizza in every building have been estimated. The buildings can be visited many times, and by different delivery men too. However, in every one of them, pizza sale can be made only once.

The spaces to let should be chosen and delivery routes should be planned in such a way that the total profit (revenues of pizza sale reduced by rent and delivery costs) is maximized.

Input

In the first line of the input there are two integers n and k ($1 \leq k \leq n \leq 2000$) which specify the number of junctions in the city and maximal number of commercial spaces available to let.

In the second line there are n integers w_1, \dots, w_n ($1 \leq w_i \leq 4 \cdot 10^5$); the number w_i specifies the rent cost of space in the building next to the i -th junction. In the third line there are n integers z_1, \dots, z_n ($1 \leq z_i \leq 4 \cdot 10^5$); the number z_i specifies the revenue from pizza sale in the building next to the i -th junction.

The next $n - 1$ lines describes Limonian road network; in the i -th line there are three integers a_i, b_i, p_i ($1 \leq a_i, b_i \leq n$, $1 \leq p_i \leq 4 \cdot 10^5$) which mean that there is a road section connecting junctions a_i and b_i and that the cost of driving it is p_i .

Output

In the only line of the output there should be one integer, which specifies the maximal possible total profit.

Example

For the input data:

```
5 2
2 1 2 9 3
4 2 5 1 2
1 2 1
1 3 2
1 4 2
4 5 2
```

the correct result is:

```
6
```

Explanation of the example: The best plan above assumes that commercial spaces will be rented in the buildings next to the junctions 2 and 3. The route of the delivery man hired in the first building runs through the junctions 2, 1 and 2. the second delivery man delivers pizza only in one building. The rent cost of commercial spaces equals $2 + 1 = 3$, delivery cost equals $2 \cdot 1 = 2$ and estimated revenue equals $4 + 2 + 5 = 11$. Eventually, it results in profit that equals $11 - 3 - 2 = 6$.

Task: PLO

Plotter



Maratona de Programação, Day 1. Available memory: 256 MB.

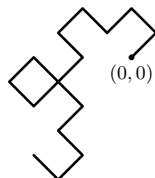
26.01.2015

To test the functionality of his newly-purchased plotter, Byteasar decided to plot a few bytecurves. A bytecurve of order n consists of 2^n segments of length $\sqrt{2}$ each. The first of them connects the points with coordinates $(0, 0)$ and $(1, 1)$. A bytecurve of order n can be described with a word L_n of length $2^n - 1$ based on a two-letter alphabet $\{L, R\}$. The word's i -th letter indicates that, after i -th segment has been printed, the pen turns and moves orthogonally to the left (letter L) or to the right (letter R) to print the next segment. In other words, the pen changes its direction to the left or to the right by 90° .

L_1 consists of a single letter L (left turn) and L_2 consists of three letters LLR, indicating two left turns followed by one right turn. Inductively, L_n is generated from L_{n-1} in the following way: separate all letters in L_{n-1} with spaces and place an additional space in front and at the end of L_{n-1} . Then, in the newly-created spaces, place alternating letters L and R, beginning with L. For example, L_3 is generated as follows:

$$L_2 = LLR \rightarrow _L_L_R_ \rightarrow \underline{LLRLRR} = L_3.$$

Similarly, $L_4 = LLRLRRLLLRRLRR$ (see the figure below).



It takes exactly one second to draw one segment of the bytecurve. While waiting for the plotter to complete printing, Byteasar ponders the following question: given a point (x, y) on the plane, when will the pen be positioned at that point? For example, for the bytecurve of order 4 in the above figure, the pen will be at point $(-3, -1)$ seven seconds after the beginning of plotting, and again eleven seconds after the beginning of plotting. Your task is to answer Byteasar's question.

Input

The first line of the input contains two integers, n and m ($1 \leq n, m \leq 2000$), where n indicates that the bytecurve is described by L_n and m indicates that there are m query points. The next m lines each contain two integers, x_i and y_i ($-10^9 \leq x_i, y_i \leq 10^9$), indicating the coordinates of the i -th query point. Note that the coordinates in any particular input line are not guaranteed to represent a point on the bytecurve. Moreover, no point appears more than once in the input.

Output

The output consists of m lines containing answers to the consecutive queries. Each answer consists of a nonnegative integer k_i , specifying the number of visits the pen makes to the query point (x_i, y_i) ; followed by k_i nonnegative integers indicating the times of those visits in increasing order, in seconds since the beginning of plotting. Numbers should be separated by single spaces, and there should be no spaces at the beginning or end of a line.

Example

For the input data:

```
4 3
-3 -1
1 1
-1 0
```

the correct result is:

```
2 7 11
1 1
0
```

Task: QRC

QR Codes



Maratona de Programação, Day 1. Available memory: 256 MB.

26.01.2015

The Mayor of Limonia is a great fan of all sorts of computer novelties. Lately he noticed that there are more and more QR codes (*Quick Response*) appearing around him. After looking into the topic, he decided to use these codes for encoding the numbers of official letters. Your team got the opportunity to prove yourselves in front of the Mayor by writing generator for such codes.

QR code is a square with side length of n , divided into $n \times n$ cells. Every cell can be either bright or dark (which carries one bit of information). Not all of the layouts of bright and dark cells are equally easily recognizable by decoding algorithm. In particular, large groups of cells of the same brightness are undesirable. Therefore, the specification of the QR codes defines the set of masks that can be applied to the generated code — such a mask changes the brightness of selected cells. In order to estimate how well certain mask works, a metric has been implemented. It penalizes undesirable layouts of cells. To determine the metric, one should, among other things, sum up all the areas of all rectangles consisting of cells of the same brightness.

It turns out not to be such an easy task. Peeping at competitors' solutions did not help either — it appears that available implementations can only give an approximate metric by counting only the rectangles with sizes of 2×2 . The Mayor does not agree to half-measures and requires full implementation of the specification. Convince him that you are ready to face this challenge!

Input

In the first line of the input there are two integers n and m ($1 \leq n \leq 2000$, $0 \leq m \leq 200\,000$) which specify the square side length and the number of dark cells. In the next m lines there are descriptions of the dark cells: in the i -th line there are two integers w_i and k_i ($1 \leq w_i, k_i \leq n$) which specify the row and the column of the i -th dark cell. Every cell appears in the input no more than once.

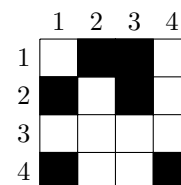
Output

The output should contain one integer that will convince the Mayor about your competence. It is supposed to be the remainder of dividing the number of the all summed up areas of one-coloured rectangles by 1 000 000 007.

Example

For the input data:

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



the correct result is:

```
58
```

Explanation of the example: The total area of dark rectangles is 10 (6 rectangles with area of 1 and 2 rectangles with area of 2), and the area of bright rectangles equals 48.

Task: WES

Funfair



Maratona de Programação, Day 1. Available memory: 256 MB.

26.01.2015

Byteasar and Byteon go to a funfair. They have been there many times and they know all the attractions, so each of them has prepared beforehand a list of his favourite attractions in order he would like to visit them. The lists differ, so the friends decide to cross out some attractions on the lists in such a way the lists become identical. They do not want to change the original ordering of the lists. Moreover, they would like the final list to be as long as possible. How many different lists they could get?

Input

In the first line of the input there are two integers n_1 and n_2 ($1 \leq n_1, n_2 \leq 10\,000$), specifying the lengths of lists of both friends. The next two lines contain the lists (of length n_1 and n_2 , respectively); each of them contains integers from range $[1, 10\,000]$ separated by single spaces. Each integer denotes one attraction at the funfair.

Output

In the first and only line of the output you have to write a single integer. It should stand for the number of different plans of visiting the funfair, which can result from crossing out some attractions on friends' lists. The answer should be given modulo 1 000 000 007.

Example

For the input data:

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

the correct result is:

```
2
```

Explanation of the example: The longest final list has length of 3, and is either 1, 2, 1 or 1, 3, 1.

Task: WOM

Wombat



Maratona de Programação, Day 1. Available memory: 256 MB.

26.01.2015

Puffnootsy the wombat is the favourite pet of Andrew Citrus — the Mayor of Limonia. Earlier it belonged to Andrew's father, who passed it to his son. The Mayor wants to build his wombat a new pen so he can perform various frolics and feel just like a real Australian wombat.

Puffnootsy is an animal that has more than a few years on the back of his neck and is no longer in its prime. In particular, he does not see well, so he uses the sense of smell for orientation. To make it easier for Puffnootsy to move inside the pen, Andrew Citrus came to the conclusion, that the pen should be as uncomplicated in shape as possible. He plans it to be a polygon, in which every two neighbour sides are perpendicular to each other. An additional convenience for the wombat will be the fact that every two sides will be of different lengths expressed in integers. Thanks to that, Puffnootsy, sniffing the wall from the beginning to the end, will learn its length, so that he will always know his current position inside the pen.

So Andrew Citrus faced a difficult task of designing the pen so that it meets requirements listed above. Moreover, he wishes it consisted of precisely n walls. Help him and suggest a design for such a pen! As pen should be built according to strict Limonian standards (thus it will be quite expensive), the Mayor will appreciate if you do your best to minimize the circumference of the pen.

Input

In the first and only line of the input there is one integer n ($6 \leq n \leq 20\,000$) specifying the number of the walls the pen should consist of. You can assume that there exists a pen with n walls that meets the problem's requirements.

Output

In the first line of the output there should be a number p specifying the circumference of the constructed pen of the smallest circumference. In the next n lines there should be the description of the constructed pen. In the i -th line there should be two integers specifying the coordinates of the i -th vertex of the pen.

The walls of the pen have to be parallel to the axes of the coordinate system. All of the vertices have to be pairwise different. The vertices printed in two consecutive lines must be neighbours on the circumference of the pen. No wall of the pen can have any common points with other walls, except for a common end of neighbouring walls. The absolute value of the vertex coordinates must not exceed 10^{12} .

Example

For the input data:

```
6
```

the correct result is:

```
22
0 0
0 7
4 7
4 5
3 5
3 0
```

Explanation of the example: The sequence of vertices above describes the pen with circumference of 22 that meets the problem's requirements.