(0,1) Matrices

Given two integers k and n, your program determines the maximum number of 2 by 2 matrices with the following constraints.

1. The matrices contain only 0s and 1s.
2. The matrices may not contain any column of all zero values.

Input Format

Two integer values n and k read from console.

Constraints

n>0 k>0

Output Format

A single integer value displayed on console.

Sample Input 0

```
2 2
```

Sample Output 0

```
9
```

Explanation 0

For the input with n=2 and k=2, we may have 9 matrices satisfying the given constraints.

$$\begin{bmatrix}1 & 0\\1 & 1\end{bmatrix} \begin{bmatrix}1 & 1\\1 & 1\end{bmatrix} \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \begin{bmatrix}1 & 1\\0 & 0\end{bmatrix} \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix} \begin{bmatrix}0 & 0\\1 & 1\end{bmatrix} \begin{bmatrix}0 & 1\\1 & 1\end{bmatrix} \begin{bmatrix}1 & 1\\0 & 1\end{bmatrix} \begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}$$

ABC Delivery Company

ABC Delivery Company is operating in a country, delivering packages between n cities. Cities are connected by a two-way road links. Not all pairs of cities have direct links between them. Truck drivers take the least cost route in going from a city A to a city B. Cost is measured in terms of the driving time. The CEO of the company says: “We guarantee delivery of a package from any city to any other city within X hours.” What is the value of X?

For example for a set of 6 cities in sample input 1, the longest trip (time-wise) takes 21 hours and it starts in city C1 and ends in city C4, passing through cities C2 and C5. Therefore X is 21 hours. Every other trip, from a city to another city, takes less than or equal to 21 hours.

Input Format

Line 1 of the input is n, the number of cities. Line 2 is m, the number of direct links between cities. Line 2 is followed by m lines – each line describing a link, in the format: cityname1 cityname2 cost. A city name is character ‘C’ followed by a city number, e.g. C0, C23, C9. Cities are numbered from 0 to n - 1.

Constraints

6<= n <= 30

and

m >= n-1.

Every city can be reached from every other city.

Output Format

For each test case, output should be the cost of the longest trip followed by in the next line, the longest trip, starting with the origin city and ending with the destination city.

Sample Input 0

```
6
8
C0 C2 9
C0 C5 14
C1 C2 10
C1 C3 15
C2 C5 2
C2 C3 11
C3 C4 6
C4 C5 9
```

Sample Output 0

```
Longest Trip Time: 21
Trip: C1 C2 C5 C4
```

Explanation 0

For example for a set of 6 cities in sample input 1, given below, the longest trip (time-wise) takes 21 hours and it starts in city C1 and ends in city C4, passing through cities C2 and C5. Therefore X is 21 hours. Every other trip, from a city to another city, takes less than or equal to 21 hours.

Ahmed and his Camel

Ahmed and Saleh are best friends. Saleh for some reasons had to go and visit his relative in a distant city. Its been a while and Ahmed has lost all contact with Saleh. Ahmed needs to go in the desert looking for Saleh. In the desert there are many wells and oasis. Ahmed knows that if he traverses through the desert he could be lost in the vastness of the desert, dehydrate and die, so he decided to number these wells and oasis as integer values starting from 0 onwards. He is recording which wells he visited. It is a possibility that Ahmed can go in loops looking for his friend.

Your program reads Ahmed log where he is recording this information and warns him if he has entered a loop.

Help Ahmed find his friend!!

Input Format

On the first line there will be an integer t, the number of test cases to follow. For each test case, there will be 2 input lines:

On the first line of the test case, there will be 2 integers n and m, where n is the number of vertices and m is the number of edges.

On the second line, there will be m pairs of integers separated by a space character. Each pair shows a two way connection between vertex a and vertex b.

Constraints

1 <=t <=1000 1 <=n <=1000 1<=m<=10000 0<=a,b<=n-1

There can be multiple edges and/or self-loops. In this case we consider the graph to contain a loop.

Output Format

For each test case you will have to write one line that contains an integer, in the case where there is a loop you will write the number 1 or else you will write the number 0.

Sample Input 0

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

Sample Output 0

```
1
```

Ali Baba’s Choices

Ali Baba is in the magical cave full of treasures of gold – crowns (c), necklaces (n), bangles (b), ingots (i), rings (r), goblets (g) and daggers (d) of different shapes and sizes – altogether there are n items. Each item has its weight marked on it:. Ali Baba’s donkey, tethered outside the cave, can carry no more than W units of weight. To maximize the total value of his picks he must choose items whose weight adds up to exactly W units of weight. And Ali Baba has his personal preferences – he wants to take at least ingots, goblets and daggers. How many choices does he have?

Input Format

First line is an integer n, n is the number of items in the cave Second line is an integer W, W is the maximum weight of the selected items. Third line contains comma seperated integers. These are the weights of n items. Fourth line is comma seperated n items as characters. Fifth line is 3 comma seperated integers. These are the minimum number of ingots, goblets and daggers to be carried.

Constraints

10<=n<=32

Output Format

Single integer value representing the number of choices.

Sample Input 0

```
10
20
8,  5,  10, 11, 5, 3, 4, 9, 7, 2
'b', 'n', 'i', 'i', 'g', 'g', 'd', 'd', 'i', 'r'
1, 1, 1
```

Sample Output 0

```
2
```

Explanation 0

There are two choices:

11 – ingot, 3 – goblet, 4 – dagger, 2 – ring
and

11 – ingot, 5 – goblet, 4 – dagger

that satisfy the constraints

Candy thief

Little Billy is looking for the candy that his mum hidden at their house and he is determined to get it all. Little Billy managed to get his hands on the floor plan of the house, and realized that the house is full of fake walls. Write a program to help Little Billy get as much candy as possible as quickly as possible.

Input Format

The input will contain two test cases. The first line of each test case is an integer 3 <= N <= 10, the represents the number of rows and columns in the floor plan, followed by N lines (each N characters long) describing the floor plan. (.) – is an empty space (#) – is an unpassable wall (B) – is Billy's start location (*) – is a piece of candy (a-f lowercase letters) - are fake walls Fake walls are unlockable only if Billy has already collected at least as much pieces of candy as the order of the letter. That is, door "a" requires at least 1 piece of candy; "b" requires 2; "f" requires 6.

Constraints

3 <= N <= 10

Output Format

The output will contain 2 lines of output, each being a pair of non-negative integers C and S. C is the maximum number of candy pieces that Billy can collect, and S is the minimum number of steps to do so.

Sample Input 0

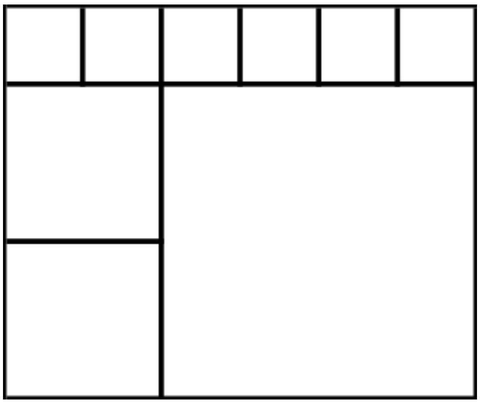
```
3
B.*
##b
..*
```

Sample Output 0

```
1 2
```

Carpeters

In a big palace, a rectangular reception hall needs to be carpeted with square carpets. The carpets come in different sizes, but they all measure in some power of two: 1, 2, 4, 8, etc. You are asked to write a program to find the minimum number of carpets needed to carpet the reception hall. For example, a 5 by 6 space can be carpeted with 30 of the smallest carpet, but the minimum number of carpets required is only 9, as indicated below.



Input Format

Each test case consists of 1 line containing 2 integers representing the length L and width W of the hall, separated by a single space.

Constraints

1 <= L <= 1000 1 <= W <= 1000

Output Format

For each test case, output one line of data containing the minimum number of carpets necessary to exactly cover the W by L hall.

Sample Input 0

```
50 40
```

Sample Output 0

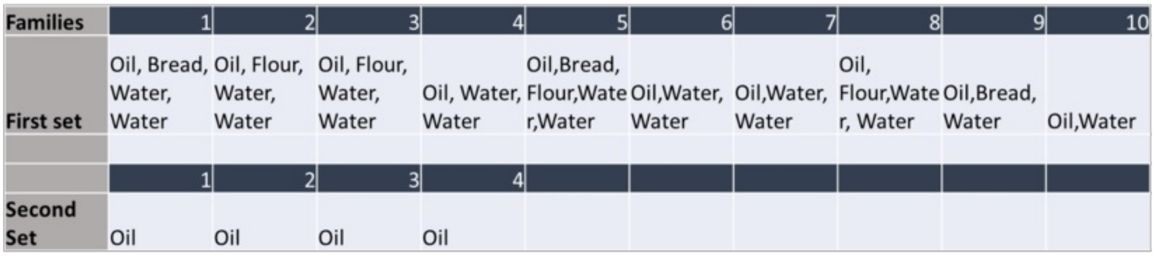
```
29
```

CrazyDonor!

You are given an authority to distribute the food in a remote area. The number of families in that war-torn area is random so you usually take more than what is required but usually you might take all items or sometime few types of item. You are given set of items like Oil, Chocolate, Bread, Flour, Jelly, Water, Jam, Butter, Rice and Blanket. You have to distribute the items with following instruction • First food (Oil) item will be given to all families • Second food item will be given only to family with number 7, 14, 21 and so on • Third food item will be given only to families with number 1, 5,9,13 and so on • Fourth food item will be given to families with number 2,3,5,8,12 and so on • Fifth food item will be given to families with no number • Sixth food item will be given to all families 2 packs • Seventh item will be given to all families with numbers 1, 2,3,5,8,13 and so on • Eight item will be given to all families with number 2, 4, 6, 8 and so on • Ninth item will be given to any family that has got either food item sixth or food item third. • Tenth item will be given to family that got food item second and seventh

Example 2 6 10 2 4

First Set 6 Food Items 10 Families Second Set 2 Food Items 4 Families



Families	1	2	3	4	5	6	7	8	9	10
First set	Oil, Bread, Water, Water	Oil, Flour, Water, Water	Oil, Flour, Water, Water	Oil, Water, Water	Oil,Bread, Flour,Water,Water	Oil,Water, Water	Oil,Water, Water	Oil, Flour,Water, Water	Oil,Bread, Water	Oil,Water
	1	2	3	4						
Second Set	Oil	Oil	Oil	Oil						

Example Input File 2 6 10 2 4

Example Output to Screen Total number of items distributed: 38 All families that have Bread are: 1 5 9 Maximum Items got by family is: 5 All families that got only one items are: 0

Total number of items distributed: 4 All families that have Bread are: 0 Maximum Items got by family is: 1 2 3 4 All families that got only one items are: 1 2 3 4

Input Format

The first line will contain a single integer n that indicates the number of data sets that follow. Each line contains two number n and m where n is number of food items present in the Truck for distribution and m is the number of families to get the food.

Constraints

Already described in problem statement

Output Format

Each data set will have five output statements, labeled, formatted and spaced as shown below: • Total number of items distributed #N • All families that Bread are (List of families) e.g. 2,10 • Maximum Items got by family is : (List of families) e.g. 2,10 • All families that got only one items are : #N

Sample Input 0

```
2
6 10
2 4
```

Sample Output 0

```
Total number of items distributed: 38
All families that have Bread are: 1 5 9
Maximum Items got by family is: 5
All families that got only one items are: 0

Total number of items distributed: 4
All families that have Bread are: 0
Maximum Items got by family is: 1 2 3 4
All families that got only one items are: 1 2 3 4
```

Explanation 0

Total number of coming inputs are 2 6 is the number of items and 10 are the number of families, rest the output is based on contraints

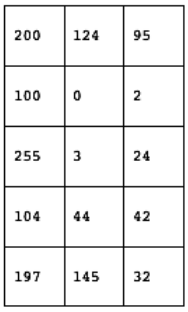
Image Pattern Classification

In image processing, pattern classification is a very important topic which consists in identifying specific patterns in an image and then analyze them.

In this problem, we aim at implementing a pattern classification algorithm in gray scale images.

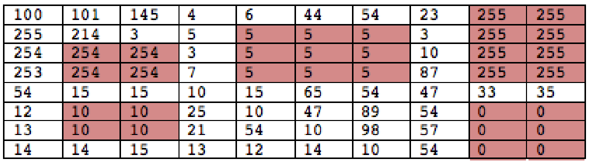
We consider a grayscale image as a two-dimensional matrix with values encoded on one byte from 0 to 255. 0 means pure black and 255 means pure white color and values in between are gray colors with different intensities. The width represents the number of column and the height represents the number of rows. Every element represents a pixel.

For example, the following matrix represents an image with width 3 column and height 5 rows.



200	124	95
100	0	2
255	3	24
104	44	42
197	145	32

In this problem, we would like to identify rectangular patterns. Let us illustrate this in the following example. Consider the following image:



100	101	145	4	6	44	54	23	255	255
255	214	3	5	5	5	5	3	255	255
254	254	254	3	5	5	5	10	255	255
253	254	254	7	5	5	5	87	255	255
54	15	15	10	15	65	54	47	33	35
12	10	10	25	10	47	89	54	0	0
13	10	10	21	54	10	98	57	0	0
14	14	15	13	12	14	10	54	0	0

In this image, we have five rectangular classes that are highlighted in red in the above example.

For such a matrix, the output must be in the following structure

```
5:[254,(2,1),(2,2)][10,(5,1),(2,2)][5,(1,4),(3,3)][255,(0,8),(4,2)][0,(5,8),(3,2)]
```

This means the following:

- 5 represent the number of classes
- 254: represents the color of the first class
- (2,1) represents the index of row and index of column of top-left pixel in the image of the color class 254
- (2,2) represents the height (number of rows) and width (number of columns) of the pattern with color class 254
- 10: represents the code of the second class
- (4,1) represents the index of row and index of column of top-left pixel in the image of the color class 10
- (2,2) represents the height (number of rows) and width (number of columns) of the pattern with color class 10

Input Format

The input will be as follows

```
100 101 145 4 6 44 54 23 255 255
255 214 3 5 5 5 5 3 255 255
254 254 254 3 5 5 5 10 255 255
253 254 254 7 5 5 5 87 255 255
54 15 15 10 15 65 54 47 33 35
12 10 10 25 10 47 89 54 0 0
13 10 10 21 54 10 98 57 0 0
14 14 15 13 12 14 10 54 0 0
```

Constraints

- There are no spaces in the output. Only separators like : [] and , are used like shown in the example.
- A rectangle must have at least 2 rows and 2 columns. Rectangles with only 1 row and/or one column are not counted.
- We assume that there are no overlapping classes with the same color.
- The output must show from the left most and up most elements going down then right direction

Output Format

```
5:[254,(2,1)][10,(5,1)][5,(1,4)][255,(0,8)][0,(5,8)]
```

Sample Input 0

```
100	101	145	4	6	44	54	23	255	255
255	214	3	5	5	5	5	3	255	255
254	254	254	3	5	5	5	10	255	255
253	254	254	7	5	5	5	87	255	255
54	15	15	10	15	65	54	47	33	35
12	10	10	25	10	47	89	54	0	0
13	10	10	21	54	10	98	57	0	0
14	14	15	13	12	14	10	54	0	0
```

Sample Output 0

```
5:[254,(2,1),(2,2)][10,(5,1),(2,2)][5,(1,4),(3,3)][255,(0,8),(4,2)][0,(5,8),(3,2)]
```

Explanation 0

- 5 represents the number of classes
- 254: represents the color of the first class
- (2,1) represents the index of row and index of column of top-left pixel in the image of the color class 254
- (2,2) represents the height (number of rows) and width (number of columns) of the pattern with color class 254
- 10: represents the code of the second class
- (4,1) represents the index of row and index of column of top-left pixel in the image of the color class 10
- (2,2) represents the height (number of rows) and width (number of columns) of the pattern with color class 10

King Hashmeput and his tax-men

King Hashmeput was the king of the ancient Egyptian civilization and ruled over the Nile basin. His kingdom was constantly at war with the Nubian empire. To prepare his army, as always, he needed to collect as much gold as possible in the shortest period of time. He ordered his geographers to come up with a scheme that will allow his tax men to visit the many villages and towns to collect gold in the shortest period of time. Unfortunately, there was no accurate means to measure the amount of gold available in a settlement.

The chief geographer at Hashmeput's court proposed a strange idea to solve the problem. He presents a map of the kingdom with all cities and villages (settlements) labeled as integer values Ki which represents the population of a settlement i. He estimates that the amount of gold Gi available at a settlement i can be estimated by a number of unique consecutive prime numbers, starting from 2 such that their product is less than or equal to Ki. (Example: A village with population 500 inhabitants will yield 4 kilos of gold. 2 * 3 * 5 * 7 = 210 <= 500)

Agreeing with his chief geographer, King Hashmeput sends his tax-men on the mission to collect all the possible gold. The mission will start from the settlement with the highest population (largest integer value) and end with settlement with lowest population. The map also shows the distance between all settlements if there is a route available. There is always a route available from the starting and finishing settlements.

Note: All population values are unique.

Input Format

The first line contains two integers n and m. n represents the number of settlements and m represents the number of connections between them. Each of the next n lines contains the population of all settlements to be visited. Each of the next m lines contains three integers starting, terminating, distance which represent the starting, terminating settlements and the distance between them.

Constraints

n >= 1 m >= 1

Output Format

One integer, the maximum kilos of gold that a mission can collect in the shortest possible time.

Sample Input 0

```
4 5
5
230
241
28
28 5 100
5 241 5000
241 28 300
230 28 200
241 230 100
```

Sample Output 0

```
11
```

Explanation 0

The tax-men will start at settlement with population 241 and end at 5 taking the shortest route 241->230->28->5. They will collect 4 + 4 + 2 + 1 = 11 kilos of gold on the way.

Lucky Shopper

ABC supermarket has an award for a lucky shopper. A lucky draw is organized yearly among all shoppers. The lucky winner walks through a giant-sized checkerboard (8 X 8) with money prizes at each of the squares on the board. The lucky shopper must start at the lower left corner of the board and move to the upper right corner, by taking steps either to the right or above (moving to the left, down or on a diagonal is not allowed). The lucky shopper claims each of the money prizes at each of the squares he stepped on. You are asked to write a program to find the three best routes through the board that yield the most money for the lucky shopper.

Input Format

Each test case consists of 8 lines, each line consisting of 8 integers, separated by a single space. These integers A represent the amount of money on each square of the board.

Constraints

0 <= A <= 99999

Output Format

For each test case, output three lines of data containing the amount of money that would be obtained on the three best routes, from best to third best.
Note: Different routes may yield the same amount

Sample Input 0

```
9 2 6 6 3 7 0 9
6 7 3 1 6 9 8 4
3 0 3 9 9 7 7 5
4 4 9 2 7 4 4 6
8 6 2 4 6 5 6 7
1 4 8 9 1 8 3 7
9 1 4 8 4 1 6 4
1 9 9 6 1 2 7 6
```

Sample Output 0

```
105
104
103
```

Parser

Your program is a reports parser that accurately parses data from the console. Your parser works with the following format:

```
<Student_id> <Family_name> <First_name> <Second_name> <Third_name> <User_name> <Floor> <Room>
```

```
214110512 ALSAYED MOHAMMAD AHMED QASIM  malsayed F01 LF01-A19
212100234 ALOTAIBI FAISAL ABDULLAH falotaibi F01 LF01-A21
214100345 BINHASHMI NASSER MOUSA AHMED binhashmi F01
213110034 alzaid M. A. alzaid F01 LF07-A19
214100712 HUSSEIN hussein F01 LF01-A19
```

Each line on the console is a record with a number of fields. Some of the field values may not be present. The values of first field (id. number), second field (the family name), sixth field (the user name) and the seventh field (the floor) are always present. The values of the remaining fields can be missing. The task is to read the value in each field and output the following information in the format shown.

```
ID: 214110512
FAMILY NAME: ALSAYED
NAME: MOHAMMAD AHMED QASIM
USER NAME: malsayed
LAB: LF01-A19
--------------
ID: 212100234
FAMILY NAME: ALOTAIBI
NAME: FAISAL ABDULLAH
USER NAME: falotaibi
LAB: LF01-A21
---------------
...
```

A missing value is indicated by three stars in the output. The value of the ‘floor’ field is same for every record (F01) and is omitted from the output.

Input Format

The input from console may have a maximum of 8 values per line and any number of lines. The input ends with an empty line. First value is a nine digit id number, the second, third, fourth and the fifth values are family name, first name, second name and third name respectively, the sixth value is the user name, the seventh value is the floor and the eighth value is the lab room number.

Constraints

Values in first, second, sixth and seventh field are present in every line (or record); values in other fields may be missing. The value of the seventh field is the same for all lines i.e. F01.

Output Format

For each line of the input, the output should consist of five lines indicating id, family name, name (consisting of all the names present, except the family name), user name and the lab room number. A missing value for any of these should be indicated by three stars

Sample Input 0

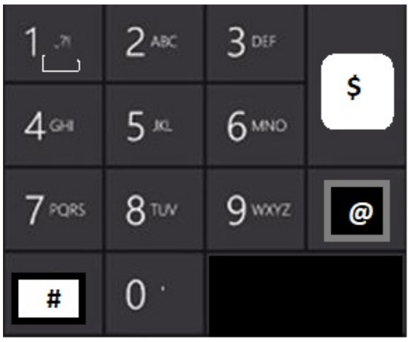
```
214110512 ALSAYED MOHAMMAD AHMED QASIM    malsayed F01 LF01-A19
214100345 BINHASHMI NASSER MOUSA AHMED    binhashmi F01
```

Sample Output 0

```
ID: 214110512
FAMILY NAME: ALSAYED
NAME: MOHAMMAD AHMED QASIM
USER NAME: malsayed
LAB: LF01-A19
---------------
ID: 214100345
FAMILY NAME: BINHASHMI
NAME: NASSER MOUSA AHMED
USER NAME: binhashmi
LAB: ***
---------------
```

Privacy Dialer

At the 3rd Programming Jam contest, students created a secret coded mechanism using WhatsApps to communicate to each other. This was done in order to hide their smart phones from the proctors and use the older Nokia 3310 (brick) phones. They were able to write a simple WhatsApp like application than ran on the old phone with alphanumeric keypad similar to the one shown in the figure.



As the international standard follows, each numeric value on the keypad has associated English language characters. In order to type A, a user would press 2 once. To type B, a user would press 2 twice. To get C, a user types 2 three times. Whenever a person finishes data entry a hyphen character '-' is entered. The space character translates to keystroke number 1.

To be sure that nobody is eavesdropping on their communication using the older phones, they devised a mechanism "Privacy Mode" to encode data entry as follows. The values of the keypad are reverse so that:

9 corresponds to ABC 8 corresponds to DEF 7 corresponds to GHI

and so on.

You program reads data from the console and translated the numeric code to a text code.

Input Format

The first line contains one character dollar sign or hash #. The '$' indicates that the input is using the "Privacy Mode". The '#' indicates the "Normal Mode".

The second line contains a set of '#$' symbol tuples indicating the number of lines to be processed.

The rest of the lines contain the key strokes.

Constraints

If the input is invalid the program prints INVALID

Maximum number of lines to be input per run is 5. If the input exceeds five lines, your program prints OVERFLOW

Output Format

A set of valid output character Strings

Sample Input 0

```
$
#$#$
4-4444-33-0-6-9-5
222-555-33
```

Sample Output 0

```
PSU.JAM
YOU
```

Sample Input 1

```
#
#$
7-7777-88-1-5-2-6
```

Sample Output 1

```
PSU JAM
```

Explanation 1

The user entered 7 once, then 7 four times, then 8 two time, 1 one-time, 5 one-time, 2 once and 6 once so the number sent will be 7-7777-88-1-5-2-6.

This translates to

PSU JAM

PSolutions

From the console, your program reads two integer values k and n such that n >= k. n is the sum of k numbers satisfying the following constraints:

$$n = x_1 + x_2 + x_3 + \cdots + x_k \qquad where\ x_1 \geq x_2 \geq x_3 \geq \cdots \geq x_k \geq 1$$

For any values of k and n. determine P such that P is the count of all possible solutions.

Input Format

Two non negative integers k and n.

Constraints

n>0

k>0

n>k

P=0 if k>=0

Output Format

Single integer P.

Sample Input 0

```
3 7
```

Sample Output 0

```
4
```

Explanation 0

k is 3

n is 7

There are 4 possibilities:

5 + 1 + 1 =7

4 + 2 + 1 =7

3 + 3 + 1 =7

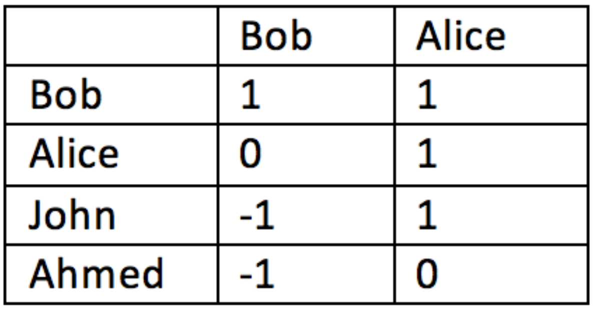
3 + 2 + 2 =7

Hence P = 4.

Social Relationship Analyzer

Bob has been using facebook for many years; he has a habit of friending anyone who requests to be a friend. This is a dilemma, over the years, he has been in a relationship for over 5000 friends, many of them he has never communicated over the months.

He is thinking of building a fast program to filter unwanted friends based on their trustfulness. You need to help him build a fast program that would require minimum number of computations. Bob uses a strange method of defining his social relationships using a 2x2 matrix composed of trust relationships. He defines 3 states of relationships, trustful (represented by 1), so so (represented by 0) and not so trustful (represented by -1) in the matrix. These matrices may not be necessarily be square matrices (i.e. rows and columns can be of different sizes).



	Bob	Alice
Bob	1	1
Alice	0	1
John	-1	1
Ahmed	-1	0

Since he has over 5000 friends, he cannot afford to have a very large matrix containing trust parameters. To determine an optimal solution, he has decided to build many small 2x2 matrices by asking his friends to provide trust information about their friends in the same manner. Luckily his friends always provide their matrices in a particular order:

A1(a x b), A2(b x c), A3(c x d) …. Where (a x b) represent dimensions of matrices (a rows and b columns) sent by Bob’s friends A1, A2 and A3… An.

Bobs algorithm computes trust scores but multiplying all of these matrices. Unfortunately multiplying all matrices take a long time and Bob is tired of waiting. He wants you to help him optimize his matrix multiplication problem by selecting an optimal order of trust matrix multiplication. He wants to make sure that any solution you provide will give the minimum number of multiplications (cost).

The cost of multiplying two matrices with dimensions 3x4 and 4x5 = 3 * 4 * 5 = 60. In multiplication of more than 2 matrices, the order of multiplication matters in determining the optimal minimum cost.

Input Format

The first line consists of a positive integer n > 0 representing number of matrices. In n subsequent lines, each line consists of dimensions of each matrix, number of rows followed by number of columns. All values for rows and columns must be >0.

```
5
4 10
10 3
3 12
12 20
20 7
```

Constraints

All values for rows and columns must be >0.

Output Format

Integer value >0 giving minimum number of computations in the optimal solution. The program writes -1 to console if any error occurs, i.e. if any number of rows or columns <=0 or the matrices cannot be multiplied.

Sample Input 0

```
5
4 10
10 3
3 12
12 20
20 7
```

Sample Output 0

```
1344
```

Explanation 0

The first row indicates number of matrices to be read from subsequent lines Each line contains a matrix with number of rows and columns

Stack'em

You have bought a new game consisting of rectangular 3-D wooden pieces, where each piece has a length, a width and a height. Given a set of pieces, the player must create a stack as high as possible with the condition that a piece can be stacked on top of another only if the dimensions of its 2D base are each strictly less than those of the lower piece. Evidently, a piece can be rotated so that any side becomes its base. It is also permitted to use many instances of the same piece type. You are asked to write a program to find, using a given set of 3-D wooden pieces, the height of the highest stack that can be formed.

Input Format

Each test case consists of 2 lines: the first line contains the number N of wooden pieces available. The next line consists of 3 * N values representing the length L, width W and height H of each of the N pieces.

Constraints

1 <= N <= 50
1 <= L <= 20
1 <= W <= 20
1 <= H <= 20
L >= W

Output Format

For each test case, output one line of data containing the height of the highest possible stack formed.

Sample Input 0

```
1
4 2 6
```

Sample Output 0

```
8
```

Explanation 0

Two instances of the same piece were used

Sample Input 1

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

Sample Output 1

```
11
```

Sample Input 2

```
3
3 2 1 6 5 4 12 10 18
```

Sample Output 2

```
42
```

The wise network technician

Alferd has recently join a network company as a network technician. The company has acquired a network installation contract with the local school. This project is on a budget and Alfred has to minimize the cost of laying network cables. You need to write a program to help Alfred minimize his costs.

There are various computers labs of different sizes. The physical dimensions of these labs permit only certain number of computers to be installed. Alfred will install a network in each computer lab connecting all computers. Each computer in the lab has two network ports. The goal is to connect all computers in a loop such that the first computer connects to the last using the two ports on each computer.

Compute the minimum length of wire required for a computer lab so that all computers in the lab are connected.

Input Format

Your program reads from the console two integers n and m defining the dimensions of the room. The program reads n number of lines each containing m number of characters. The only permitted characters are '.' and '#'. The '.' character denotes empty space, '#' character denotes a computer. The wiring can only go in up/down/left/right directions, diagonal layout of wires is not permitted. The distance between two adjacent cells is 1 meter. The program prints -1 for an invalid input.

Constraints

0<=n<=20

0<=m<=20

Output Format

An integer value showing the length of wire needed.

Sample Input 0

```
4
4
....
#...
....
.#..
```

Sample Output 0

```
6
```

Explanation 0

This is a 4 by 4 room. There are only two computers in the room. Connecting these two computers in both ports to make a loop needs 6 meters of wire.

VertiStrings

Write a program that reads a set of strings from console and outputs these vertically on the console.

Input Format

The program reads lines of text from console. It will stop reading the lines when the hash character ‘#’ is read from the console. The character ‘#’ can appear anywhere in the input.

Constraints

All valid character. # is used as terminal for input.

All output strings must be atmost 36 characters long, padded with spaces at the end where necessary. If a string is too long, only output the first 36 characters.

Output Format

The output will be the same strings presented vertically, where the left to right order will be the same as the order of the input. There will be a column of bar (‘|’) characters at each end, and separating each string, and a row of minus (‘-’) characters (1 per column) at the beginning and end.

Sample Input 0

```
012345678901234567890123456789012345
David and Jane-s wedding, March 2002, Alexandria
Bahamas Holiday August 2001
#
```

Sample Output 0

```
-------
|0|D|B|
|1|a|a|
|2|v|h|
|3|i|a|
|4|d|m|
|5| |a|
|6|a|s|
|7|n| |
|8|d|H|
|9| |o|
|0|J|l|
|1|a|i|
|2|n|d|
|3|e|a|
|4|-|y|
|5|s| |
|6| |A|
|7|w|u|
|8|e|g|
|9|d|u|
|0|d|s|
|1|i|t|
|2|n| |
|3|g|2|
|4|,|0|
|5| |0|
|6|M|1|
|7|a| |
|8|r| |
|9|c| |
|0|h| |
|1| | |
|2|2| |
|3|0| |
|4|0| |
|5|2| |
-------
```