Constellation

Problem Description

Three characters { #, \*, . } represents a constellation of stars and galaxies in space. Each galaxy is demarcated by # characters. There can be one or many stars in a given galaxy. Stars can only be in shape of vowels { A, E, I, O, U } . A collection of \* in the shape of the vowels is a star. A star is contained in a 3x3 block. Stars cannot be overlapping. The dot(.) character denotes empty space.

Given 3xN matrix comprising of { #, \*, . } character, find the galaxy and stars within them.

Note: Please pay attention to how vowel ***A***is denoted in a 3x3 block in the examples section below.

Constraints

3 <= N <= 10^5

Input

Input consists of single integer N denoting number of columns.

Output

Output contains vowels (stars) in order of their occurrence within the given galaxy. Galaxy itself is represented by # character.

Time Limit

1

Examples

Example 1

Input

18

\* . \* # \* \* \* # \* \* \* # \* \* \* . \* .

\* . \* # \* . \* # . \* . # \* \* \* \* \* \*

\* \* \* # \* \* \* # \* \* \* # \* \* \* \* . \*

Output

U#O#I#EA

Explanation

As it can be seen that the stars make the image of the alphabets U, O, I, E and A respectively.



Example 2

Input

12

\* . \* # . \* \* \* # . \* .

\* . \* # . . \* . # \* \* \*

\* \* \* # . \* \* \* # \* . \*

Output

U#I#A

Explanation

As it can be seen that the stars make the image of the alphabet U, I and A.



Closing Value

Problem Description

Stock Exchange of country XYZ is still working on pen paper mode, wherein the traders have to bid the price for buying and selling the stocks and the stocks prices are checked manually, then the buying and selling data is validated and if the condition matches then it is recorded in the record book. For example, if A wants to buy stocks at 100 and B is willing to sell the required stocks at 95, then A can buy his desired share at 95 and the price of the share will become 95. The price of shares of a company is determined by the latest transaction recorded in the record book.

As the number of transactions are increasing it is getting hard to match and record the transactions manually.

The stock exchange wants to go online and has hired Karim to make the process online. The stock exchange personnel will give him the bid and he has to design a program to match the values and if the transaction is done, then record it in the record book which will also be online.

Help Karim in recording the completed transactions and enter the closing price in record book. He has to give the closing values of all the companies whose transactions have been recorded in ascending order.

In case, if more than one matching bids are received then the trader with lower TraderId will get the preference. In case only a partial match is received then it will be a split transaction. For example, consider following transaction below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TraderId** | **TradeType** | **StockName** | **Price** | **Quantity** |
| 1 | Buy | ABC | 50 | 90 |
| 2 | Buy | ABC | 50 | 90 |
| 3 | Sell | ABC | 50 | 100 |

Trader 3 will sell his 90 stocks to Trader 1 at price of 50. Since Trader 3 has 10 stocks left, the remaining stocks can be purchased by Trader 2. Hence, Trader 1 and Trader 3 have squared off their positions whereas as Trader 2 has open bid for 80 stocks of ABC at a price of 5.

Here, Trader 3 had a split transaction which is closed whereas Trader 2 also had split transaction which is still open.

Constraints

0 < N <=10000

Input

First line contains an integer N, denoting the number of bids

Next N lines, each contains 5 space separated values denoting <TraderId, Tradetype, StockName, Price, Quantity>

Where,

TraderId is single integer containing the id of trader, smaller trader id indicates the timestamp of the trade

Tradetype can be either Buy/Sell

StockName is the name of the company that trader wants to bid

Price indicates the price at which the trader has bid

Quantity indicates the number of stocks trader wants to purchase/buy

Output

Lexicographically ascending order of Stock Name(S) along with respective values(C) in the format (S:C). If no company has traded the stock, then print "Stocks not traded"

Time Limit

1

Examples

Example 1

Input

3

1 Sell ABC 1876 173

2 Sell DEF 7160 221

3 Buy ABC 6986 864

Output

ABC:1876

Explanation

Here transaction of ABC is recorded but the DEF is not recorded as no one has bid to buy the stocks of DEF.

Trader with id 3 will buy 173 shares from trader with id 1 of Stock ABC at unit price of 1876. Since this becomes the closing price of ABC, output is ABC:1876

Example 2

Input

3

1 Sell ABC 1876 173

2 Sell DEF 7160 221

3 Buy ABC 1200 864

Output

Stocks not traded

Explanation

Since, no transaction has taken place as the buy bid is less than sell for company ABC and there is no trade to buy DEF, there is no record in the record book. Hence "Stocks not traded" will be the output.

## Railway Station

### Problem Description

Given schedule of trains and their stoppage time at a Railway Station, find minimum number of platforms needed.

Note -

If Train A's departure time is x and Train B's arrival time is x, then we can't accommodate Train B on the same platform as Train A.

### Constraints

1 <= N <= 10^5

0 <= a <= 86400

0 < b <= 86400

Number of platforms > 0

### Input

First line contains N denoting number of trains.

Next N line contain 2 integers, a and b, denoting the arrival time and stoppage time of train.

### Output

Single integer denoting the minimum numbers of platforms needed to accommodate every train.

### Time Limit

1

### Examples

Example 1

Input

3

10 2

5 10

13 5

Output

2

Explanation

The earliest arriving train at time t = 5 will arrive at platform# 1. Since it will stay there till t = 15, train arriving at time t = 10 will arrive at platform# 2. Since it will depart at time t = 12, train arriving at time t = 13 will arrive at platform# 2.

Example 2

Input

2

2 4

6 2

Output

2

Explanation

Platform #1 can accommodate train 1.

Platform #2 can accommodate train 2.

Note that the departure of train 1 is same as arrival of train 2, i.e. 6, and thus we need a separate platform to accommodate train 2.

## Minimize The Sum

### Problem Description

Given an array of integers, perform atmost K operations so that the sum of elements of final array is minimum. An operation is defined as follows -

Consider any 1 element from the array, arr[i].

Replace arr[i] by floor(arr[i]/2).

Perform next operations on updated array.

The task is to minimize the sum after atmost K operations.

### Constraints

1 <= N, K <= 10^5.

### Input

First line contains two integers N and K representing size of array and maximum numbers of operations that can be performed on the array respectively.

Second line contains N space separated integers denoting the elements of the array, arr.

### Output

Print a single integer denoting the minimum sum of the final array.

### Time Limit

1

### Examples

Example 1

Input

4 3

20 7 5 4

Output

17

Explanation

Operation 1 -> Select 20. Replace it by 10.

New array = [10, 7, 5, 4]

Operation 2 -> Select 10. Replace it by 5.

New array = [5, 7, 5, 4].

Operation 3 -> Select 7. Replace it by 3.

New array = [5,3,5,4].

Sum = 17.

## Paste Reduction

### Problem Description

Some keys in Codu's computer keyboard are not working. Fortunately for Codu, these characters are present in a previously existing text file.

He wants to write a paragraph which involves typing those characters whose keys are defunct in Codu's keyboard. Only option left for Codu is to copy-paste those characters from the previously existing text files. However, copy-pasting keys is a laborious operation since one has to switch windows and also previously copied items are lost once a new set of characters are copied. Hence, Codu wants to minimize the number of times he needs to copy-paste from that text file. Fortunately there can be situations where previously copied characters are readily available for pasting. Help Codu devise a method to minimize the number of times a paste operation is needed, given - the text he intends to type and the faulty keys

### Constraints

0 < Length of paragraph <= 1000 characters

0 < number of faulty keys in keyboard <= 36

Only a to z and 0 to 9 keys can be faulty.

Input paragraph will not contain any upper-case letter.

### Input

First line contains a paragraph that is be to be written.

Second line contains a string. This string has to be interpreted in the following fashion

All characters in that string correspond to faulty keys

That string is available for copy as-is from the other text file that Codu is referring

Also, individual characters can always be copied from the same text file

### Output

Single integer denoting minimum number of copy operations required

### Time Limit

1

### Examples

Example 1

Input

supreme court is the highest judicial court

su

Output

4

Explanation

Codu will first paste su from the file when typing characters su in the word supreme.

In the second instance, Codu will need to copy character u and paste it when typing character u in the word court.

In the third instance, Codu will need to copy character su and paste su when typing character s in the word is. Codu will back track using the left arrow key and type the characters "the highe".

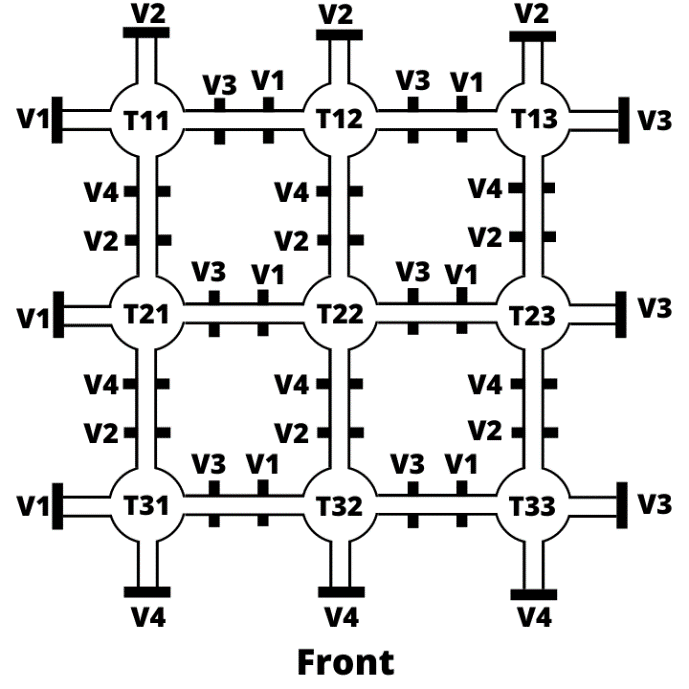
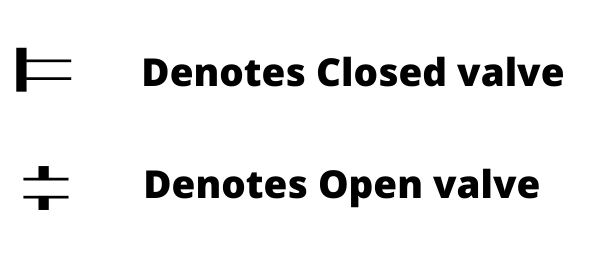
In the fourth instance, Codu will again paste su. The overall string typed until this moment is "supreme court is the highesuu". Codu will then back track again using left arrow key and type characters "t j". At this point the typed string is "supreme court is the highest juu". Cursor is after character j. Codu will use right arrow key and now the cursor will be after ju. Codu will now type "udicial co". String typed till this point is "supreme court is the highest judicial cou". Cursor is after character o. Codu will use right arrow and the cursor will be after character u. Finally Codu will type "rt".

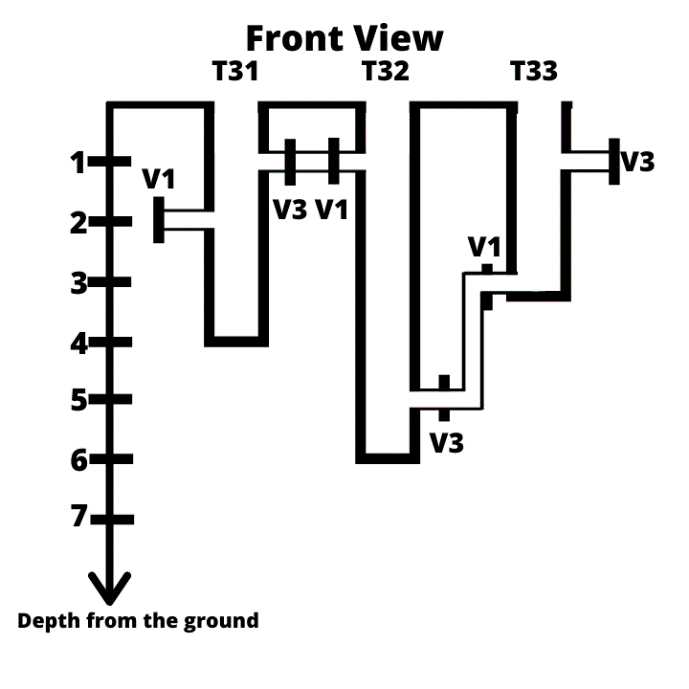
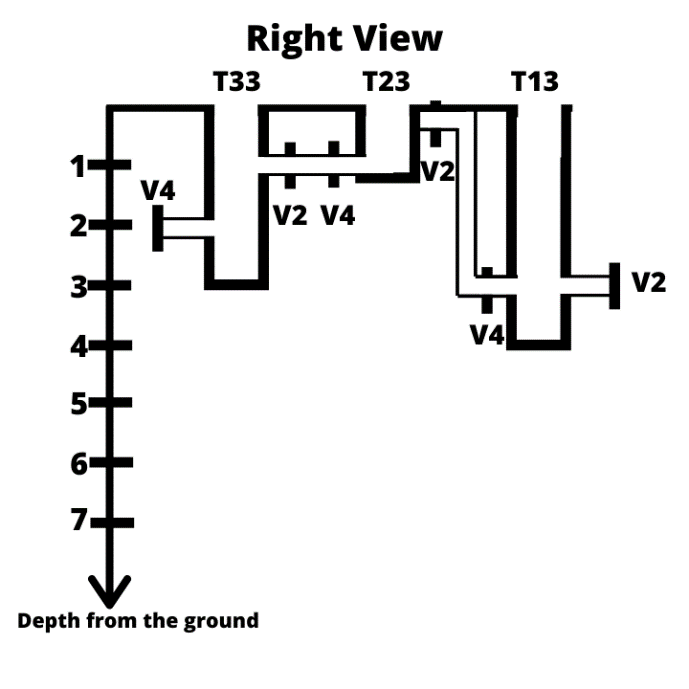
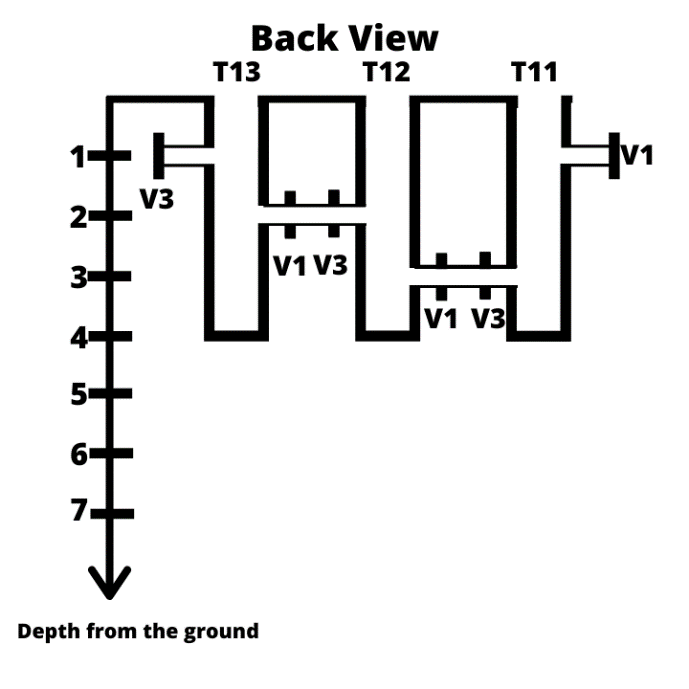
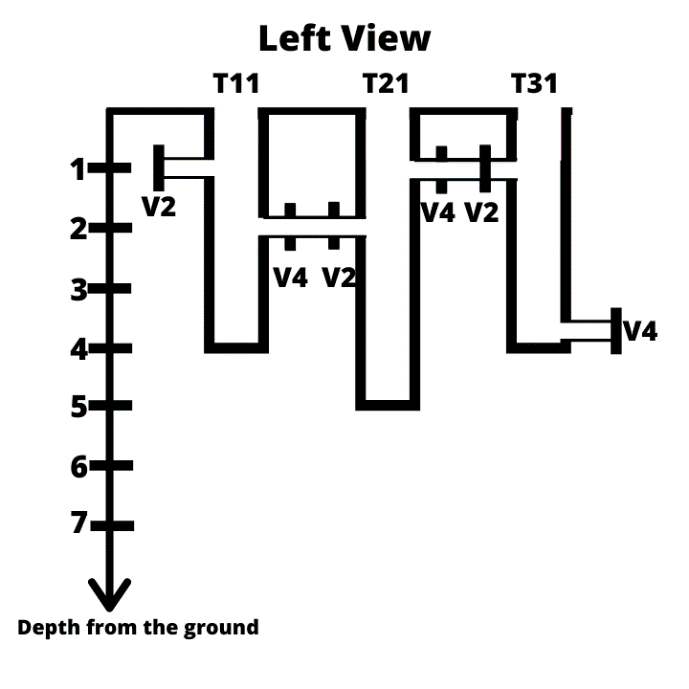
Final string - "supreme court is the highest judicial court" - is thus fully typed. Here, the number of paste operations are 4.

Tank Network

Problem Description

There are M\*N underground water tanks, which are arranged in M rows and N columns. Tanks are labelled as Tij, where i is the row number and j is the column number (1 <= i <= M and 1 <= j <= N). Each tank is of certain depth beginning from the ground level (measured in feet). Each tank has four valves situated at a certain depth from ground level. All 4 valves can be at different depth from the ground level. Valves are orthogonal to each other. Valves of each tank are labelled as V1, V2, V3 and V4. Valves may be open or closed. If a valve is closed, no water flows out through it. If a valve is open, water flows out through it. Each valve is connected to the valve of nearby tank using connection tubes (or it is terminated if there is no tank nearby). All tanks are of same shape. For all tanks, 1 liter water occupy 1 foot height of the tank. For example, below diagram depicts a 3\*3 tank network with all tanks empty.

When water starts filling into the tank, the valves attached to different tanks will determine the flow of water within the tank network.

You need to find the amount of water (in liters) required to fill a given tank (say, Txy - xth row and yth column) to a particular height h, given details about tank dimensions. You can ignore the amount of water getting filled in valves and connection tubes.

Constraints

1 <= m <= 350

1 <= n <= 350

1 <= depth of tank <= 100

0 <= position of valve <= 100

Input

First line contains 2 space separated integers denoting M and N.

Next M lines, each contains N space separated integers forming an M\*N matrix denoting the depth (in feet) of each tank.

First integer of first line in this M\*N block denotes the depth of Tank11, second integer denotes the depth of Tank12...

First integer of second line in this M\*N block denotes the depth of Tank21, second integer denotes the depth of Tank 22...

.

.

First integer of Mth line in this M\*N block denotes the depth of TankM1, second integer denotes the depth of TankM2...

Next M lines, each contains N-space separated integers forming an M\*N matrix denoting the depth (in feet) at which valve V1 of each tank is situated.

First integer of first line in this M\*N block denotes the depth of valve V1 for Tank11, second integer denotes the depth of valve V1 for Tank12...

First integer of second line in this M\*N block denotes the depth of valve V1 for Tank21, second integer denotes the depth of valve V1 for Tank22...

.

.

First integer of Mth line in this M\*N block denotes the depth of valve V1 for TankM1, second integer denotes the depth of valve V1 for TankM2...

Next M lines, each contains N space separated integers forming an M\*N matrix denoting the depth (in feet) at which valve V2 of each tank is situated.

First integer of first line in this M\*N block denotes the depth of valve V2 for Tank11, second integer denotes the depth of valve V2 for Tank12...

First integer of second line in this M\*N block denotes the depth of valve V2 for Tank21, second integer denotes the depth of valve V2 for Tank22...

.

.

First integer of Mth line in this M\*N block denotes the depth of valve V2 for TankM1, second integer denotes the depth of valve V2 for TankM2...

Next M lines, each contains N space separated integers forming an M\*N matrix denoting the depth (in feet) at which valve V3 of each tank is situated.

First integer of first line in this M\*N block denotes the depth of valve V3 for Tank11, second integer denotes the depth of valve V3 for Tank12...

First integer of second line in this M\*N block denotes the depth of valve V3 for Tank21, second integer denotes the depth of valve V3 for Tank22...

.

.

First integer of Mth line in this M\*N block denotes the depth of valve V3 for TankM1, second integer denotes the depth of valve V3 for TankM2...

Next M lines, each contains N-space separated integers forming an M\*N matrix denoting the depth (in feet) at which valve V4 of each tank is situated.

First integer of first line in this M\*N block denotes the depth of valve V4 for Tank11, second integer denotes the depth of valve V4 for Tank12...

First integer of second line in this M\*N block denotes the depth of valve V4 for Tank21, second integer denotes the depth of valve V4 for Tank22...

.

.

First integer of Mth line in this M\*N block denotes the depth of valve V4 for TankM1, second integer denotes the depth of valve V4 for TankM2...

Next M lines, each contain N space separated integers forming an M\*N matrix denoting whether the status of valve V1 i.e. { open (0), closed (1) }.

First integer of first line in this M\*N block denotes the status of valve V1 for Tank11, second integer denotes the status of valve V1 for Tank12...

First integer of second line in this M\*N block denotes the status of valve V1 for Tank21, second integer denotes the status of valve V1 for Tank22...

.

.

First integer of Mth line in this M\*N block denotes the status of valve V1 for TankM1, second integer denotes the status of valve V1 for TankM2...

Next M lines, each contain N space separated integers forming an M\*N matrix denoting whether the status of valve V2 i.e. { open (0), closed (1) }.

First integer of first line in this M\*N block denotes the status of valve V2 for Tank11, second integer denotes the status of valve V2 for Tank12...

First integer of second line in this M\*N block denotes the status of valve V2 for Tank21, second integer denotes the status of valve V2 for Tank22...

.

.

First integer of Mth line in this M\*N block denotes the status of valve V2 for TankM1, second integer denotes the status of valve V2 for TankM2...

Next M lines, each contain N space separated integers forming an M\*N matrix denoting whether the status of valve V3 i.e. { open (0), closed (1) }.

First integer of first line in this M\*N block denotes the status of valve V3 for Tank11, second integer denotes the status of valve V3 for Tank12...

First integer of second line in this M\*N block denotes the status of valve V3 for Tank21, second integer denotes the status of valve V3 for Tank22...

.

.

First integer of Mth line in this M\*N block denotes the status of valve V3 for TankM1, second integer denotes the status of valve V3 for TankM2...

Next M lines, each contain N space separated integers forming an M\*N matrix denoting whether the status of valve V1 i.e. { open (0), closed (1) }.

First integer of first line in this M\*N block denotes the status of valve V4 for Tank11, second integer denotes the status of valve V4 for Tank12...

First integer of second line in this M\*N block denotes the status of valve V4 for Tank21, second integer denotes the status of valve V4 for Tank22...

.

.

First integer of Mth line in this M\*N block denotes the status of valve V4 for TankM1, second integer denotes the status of valve V4 for TankM2...

Next line contain space separated 3 integers denoting x, y and h i.e. tank Txy filled to depth h

Output

Single line containing an integer denoting the amount of water in liters.

Time Limit

1

Examples

Example 1

Input

2 2

3 4

4 3

1 2

2 3

1 3

2 1

3 1

1 1

2 3

4 2

1 0

1 0

1 1

1 0

0 1

0 1

0 0

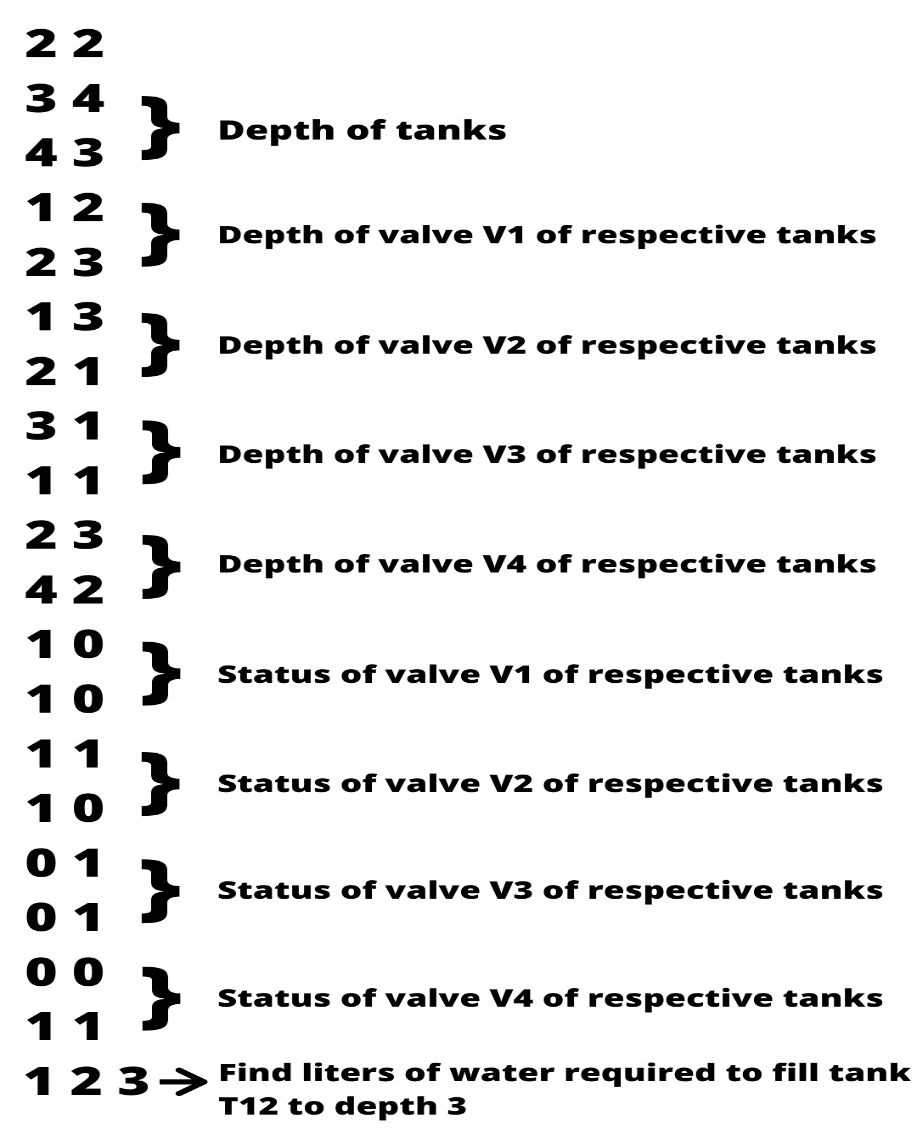
1 1

1 2 3

Output

5

Explanation :



From first input line it's clear that there are four tanks - T11, T12, T21, Tank 22, arranged in 2 rows and 2 columns.

From input lines 2 and 3, its clear that the depths of T11 is 3 feet, T12 is 4 feet, T21 is 4 feet and Tank 22 is 3 feet.

From input lines 4 and 5, its clear that valve V1 of T11 is located at a depth of 1 foot, that of T12 is at 2 feet, that of T21 is at 2 feet, and that of Tank 22 is at 3 foot.

From input lines 6 and 7, its clear that valve V2 of T11 is located at a depth of 1 foot, that of T12 is at 3 feet, that of T21 is at 2 feet, and that of Tank 22 is at 1 foot.

From input lines 8 and 9, its clear that valve V3 of T11 is located at a depth of 3 feet, that of T12 is at 1 feet, that of T21 is at 1 foot, and that of Tank 22 is at 1 foot.

From input lines 10 and 11, its clear that valve V4 of T11 is located at a depth of 2 feet, that of T12 is at 3 feet, that of T21 is at 4 foot, and that of Tank 22 is at 2 foot.

From input lines 12 and 13, its clear that the valve V1 of T11 is closed, that of T12 is opened, that of T21 is closed, and that of Tank 22 is opened.

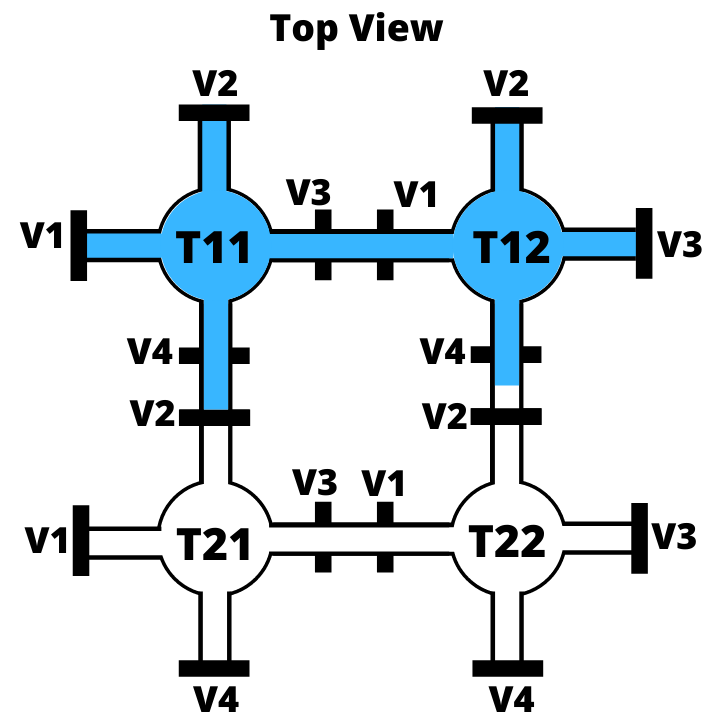
From input lines 14 and 15, its clear that the valve V2 of T11 is closed, that of T12 is closed, that of T21 is closed, and that of Tank 22 is opened.

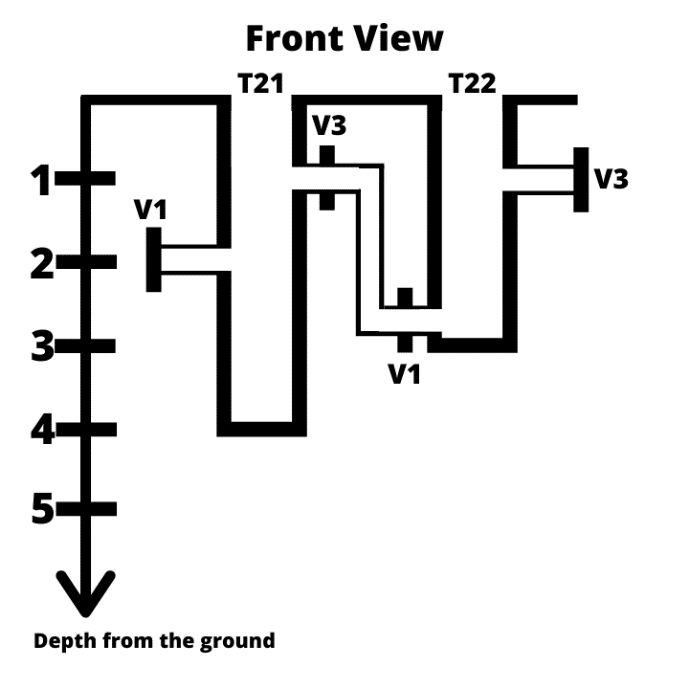
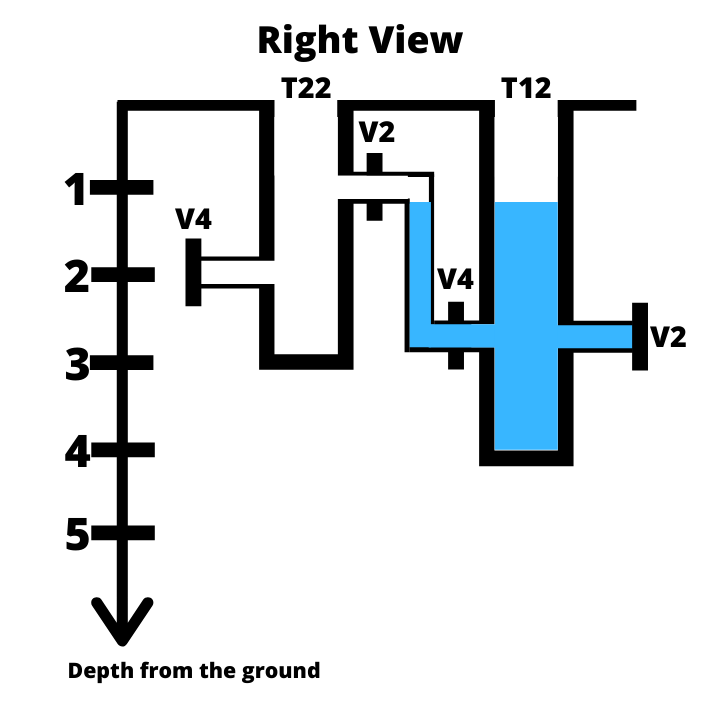
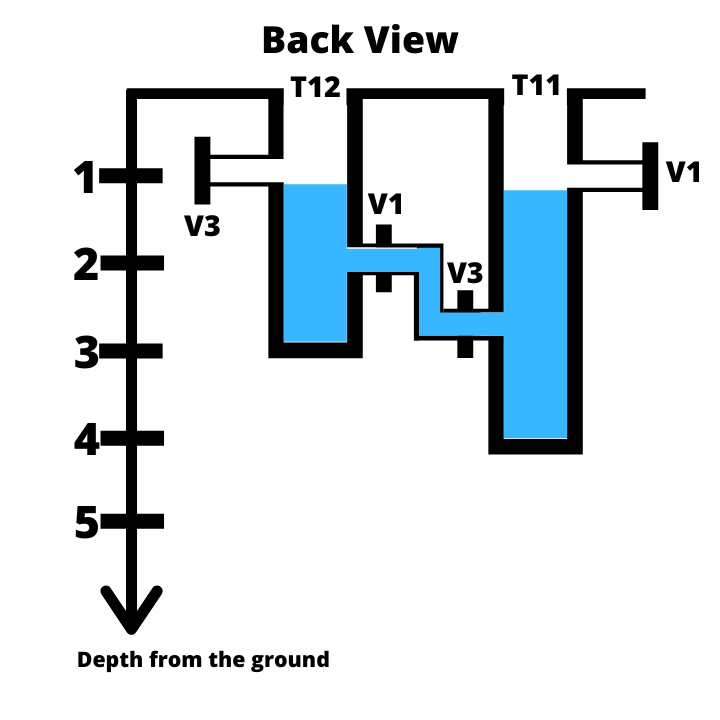
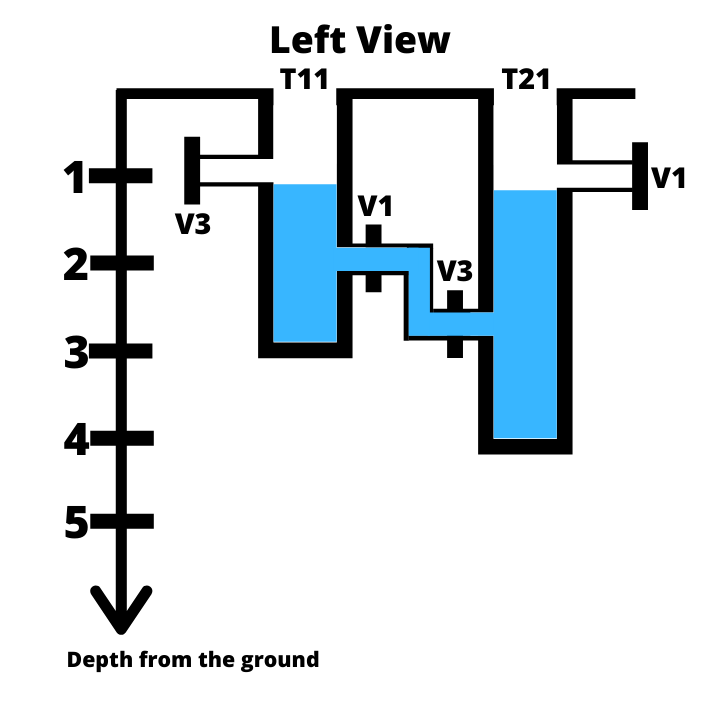
From input lines 16 and 17, its clear that the valve V3 of T11 is opened, that of T12 is closed, that of T21 is opened, and that of Tank 22 is closed.

From input lines 18 and 19, its clear that the valve V4 of T11 is opened, that of T12 is opened, that of T21 is closed, and that of Tank 22 is closed.

From input line 20, its clear that we need to find the amount of water required to fill the tank T12 for a height of 3 feet.

After filling the tank T12 to 3 feet, the tank network might look like below figure (shaded colour denotes water).



From the above diagram, it's clear that tanks T12 and T11 got filled with 3 and 2 liters of water respectively. So, total 5 liters of water is required to fill the tank T12 to a height of 3 feet.

Example 2

Input

3 3

6 6 6

6 6 6

6 6 6

1 1 1

1 1 1

1 1 1

1 1 1

1 1 1

1 1 1

1 1 1

1 1 1

1 1 1

1 1 1

1 1 1

1 1 1

1 0 0

1 0 0

1 0 0

1 1 1

0 0 0

0 0 0

0 0 1

0 0 1

0 0 1

0 0 0

0 0 0

1 1 1

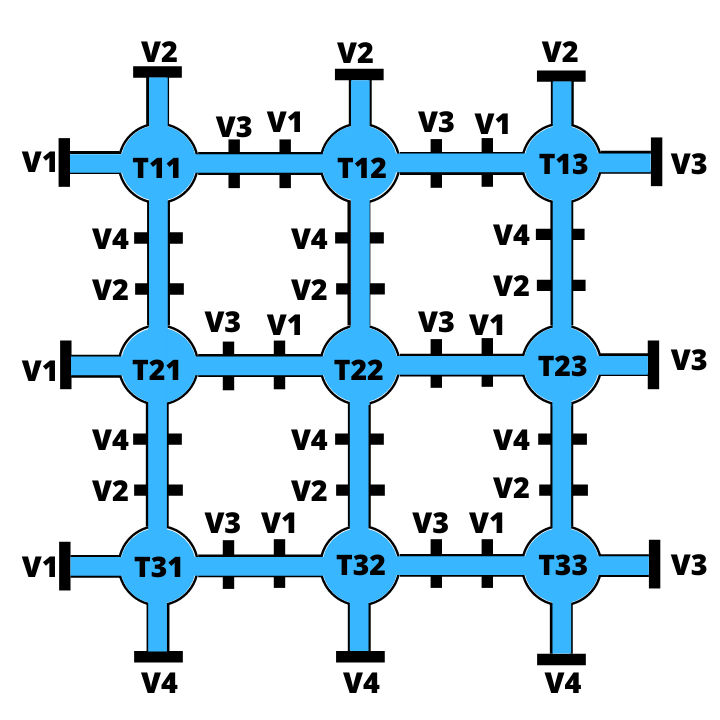
1 2 6

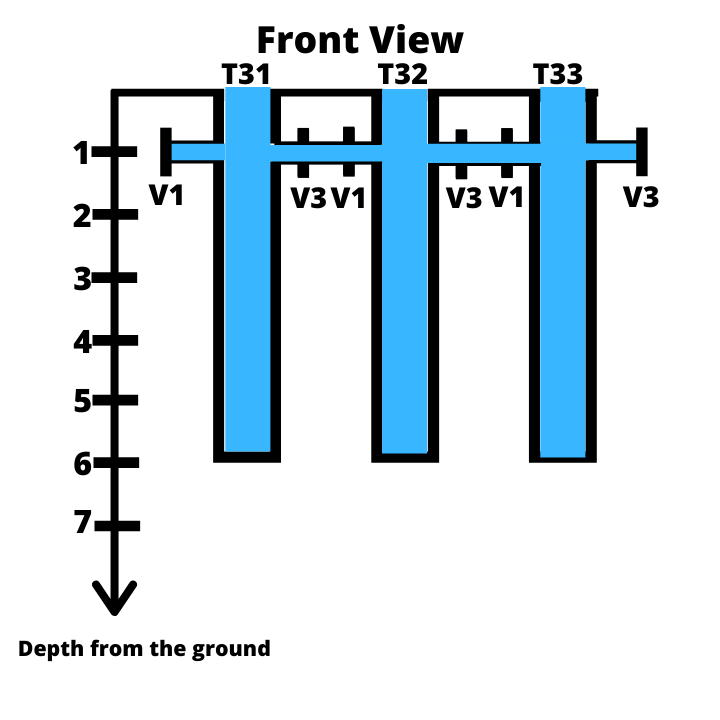
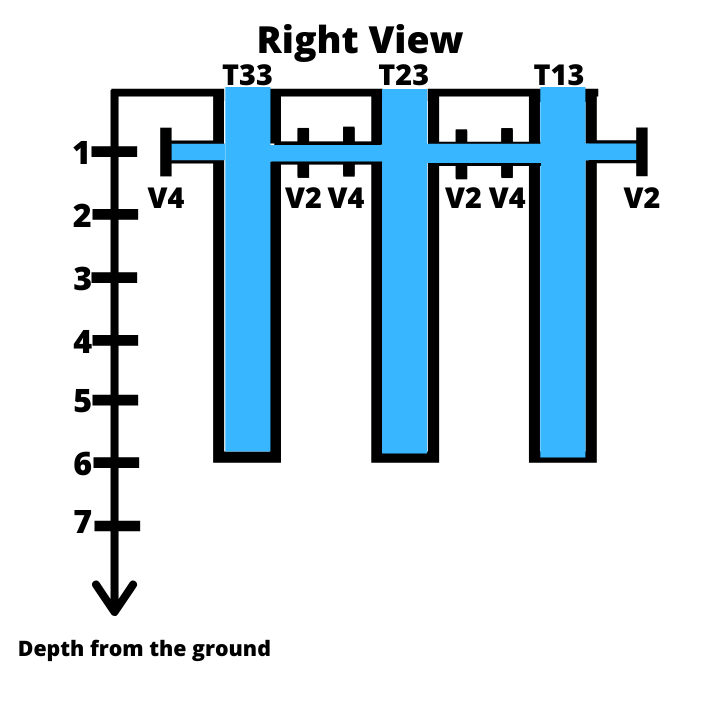
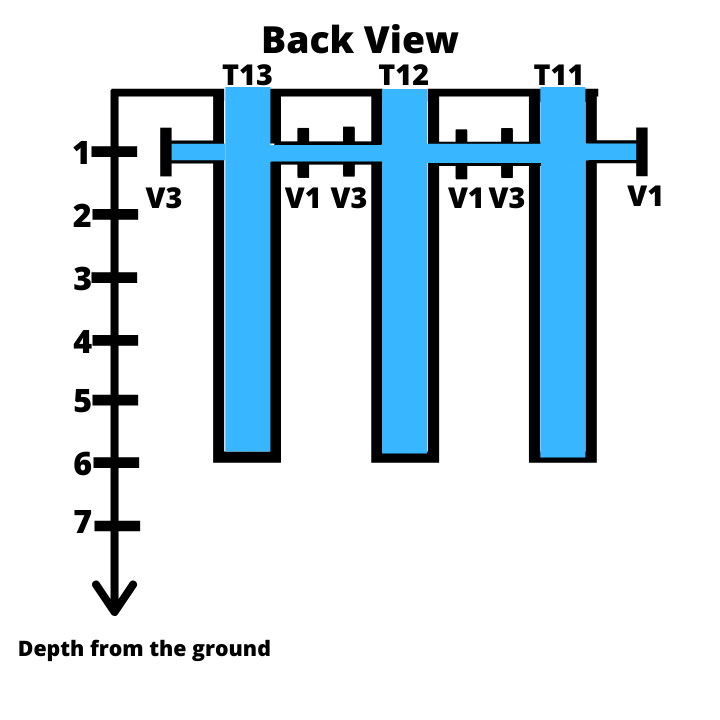
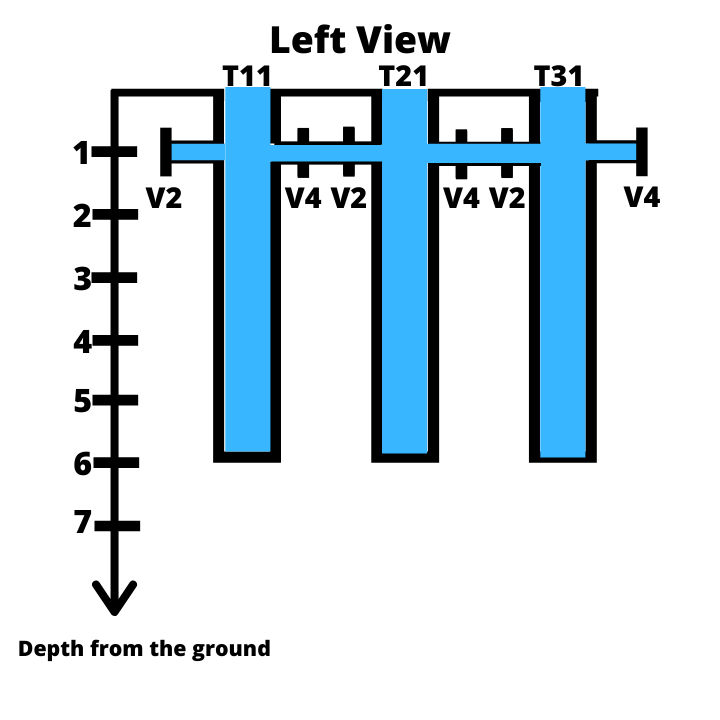
Output

54

Explanation :

After filling the tank T12 to 6 feet, the tank network might look like below figure



From the above diagram, it's clear that all tanks got filled with 6 liters of water. So, total 54 liters of water is required to fill the tank T12 to a height of 6 feet.