# Problem A. Bridge Construction

| | |
|---|---|
| Input file: | standard input |
| Output file: | standard output |
| Balloon Color: | Red |

Due to the Overpopulation our government decided to construct more bridges that can help to reduce the traffic jams. Therefore, it has decided to ask $M$ construction companies to help in these bridges. In order to construct a bridge, It must distribute the bridge into $N$ sections. Each construction company $C$ can work on sections between ($L$ and $R$) inclusive of the bridge.

The bridge is constructed if and only if all its sections have been worked on by at least one company. It should be mentioned that, the cost of building any of the bridge's sections is the number of companies that have worked on that section. Therefore, the construction cost of the whole bridge is the summation of all of its sections cost.

You are asked to choose a subset of companies that can construct the whole bridge with the minimal cost. If the bridge can not be constructed print $-1$.

## Input

First line will contain an integer $T$ the number of test cases.

Next $T$ lines each contains two integers $N$ and $M$ where $(1 \leq N, M \leq 10^5)$, followed by $M$ lines each contains two integers $L_i$ and $R_i$ where $(1 \leq L_i \leq R_i \leq N)$.

## Output

Print $T$ Lines the answer required above.

## Examples

| standard input | standard output |
|---|---|
| 1<br>4 2<br>1 3<br>2 4 | 6 |
| 1<br>4 3<br>1 1<br>2 3<br>3 3 | -1 |

# Problem B. Primes Every Where Easy Version ✓

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Black`

Given an array $a$ of size $n$.

Print the number of pairs $(i, j)$ such that $(a_i + a_j)$ is prime and $i \neq j$.

**Prime numbers are positive integers that have exactly two distinct positive divisors.**

## Input

The first line contains an integer $T$ the number of testcases.

The next $T$ line each contains an integer $n$ denoting the length of the array followed by a line contains $n$ integers $a_1, a_2, \ldots, a_n$.

- $1 \leq n \leq 100$
- $1 \leq a_i \leq 10^7$

## Output

Print $T$ lines the answer required above.

## Examples

| standard input | standard output |
|---|---|
| 1<br>5<br>2 3 5 7 2 | 8 |
| 1<br>3<br>2 3 9 | 4 |

## Note

The first test : (1,2),(2,1),(1,3),(3,1),(2,5),(5,2),(3,5),(5,3).

The second test : (1,2),(2,1),(1,3),(3,1).

# Problem C. Bandwidth Checker

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Light Blue`

We are building a website that offers a reliability and rapid response for any request. Our website is offering an online store where you can buy and sell goods.

It is worth mentioning that our testers always want to check the response of the website since it's our website advantage. The website has a bandwidth between N and M inclusive and the tester has a bandwidth X. The website will response if the sender bandwidth is between the website bandwidth inclusively.

Now, you are asked to check if our website will response to the tester or not.

## Input

Input consists of one line, containing $N, X$ and $M$ Where $(-10^9 \leq N, X, M \leq 10^9)$.

## Output

Output 'Yes' if the website reponed to the tester and print 'No' otherwise.

## Examples

| standard input | standard output |
|---|---|
| -10 0 10 | Yes |
| 1 5 3 | No |

~~DA~~ O, N, ~~AM~~ M, E, F, G

# Problem D. Lucky Range

| | |
|---|---|
| Input file: | standard input |
| Output file: | standard output |
| Balloon Color: | Purple |

Mathematics is one of the most important fields in our schools. We are training the superior students to compete in all over the world. In order to train those students, our coaches make a huge number of problems $R$. They have followed some intelligent patterns to sort the problems according to the problem's hardness.

The only thing that has been announced by our coaches is the easiest problem you can solve is the problem with the most minimum index and it has the minimum lucky among all the problems.

Note: The number's lucky is calculated as the product of its digits. As example: the number 123 has lucky equal $1 * 2 * 3 = 6$.

The students have solved $L - 1$ problems. Help them determine the next easiest problem they can solve in the range between $L$ and $R$ inclusive.

## Input

The first line contains an integer $T$ the number of test cases. each of the next $T$ lines contains two integers $L$ and $R$ where $(1 \leq L \leq R \leq 10^{18})$.

## Output

For each test case: Print a single line that contains the index's lucky of the easiest problem.

## Example

| standard input | standard output |
|---|---|
| 1<br>12111 12115 | 2 |

## Note

First Example:

The problem with index 12111 has the lucky equal $1 * 2 * 1 * 1 * 1 = 2$.

The problem with index 12112 has the lucky equal $1 * 2 * 1 * 1 * 2 = 4$.

The problem with index 12113 has the lucky equal $1 * 2 * 1 * 1 * 3 = 6$.

The problem with index 12114 has the lucky equal $1 * 2 * 1 * 1 * 4 = 8$.

The problem with index 12115 has the lucky equal $1 * 2 * 1 * 1 * 5 = 10$.

Therefore, the index of the easiest problem among them is 12111 and its lucky is 2.

1999

2999

# Problem E. Fancy Driver

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Balloon Color: | `Dark Green` |

We are building a new software for our drivers. The software helps our drivers to know the **pickup** position, **drop** position and **tips** donated by the customer who will take the ride. It should be mentioned that, for each 1KM travelled our drivers charge their passengers 1 unit of money per each. In addition to that, the driver car can accommodate only **one** passenger at any point of time.

**Mori** is one of our clever smart drivers. He always wants to **maximize** his profit that he earns by the end of the day. Unfortunately, due to the long ride he forget the amount he has earned. Therefore, he is seeking your help. **Mori** starts his day at position 0. He will give you all the $N$ passengers **pickup** position $P$, **drop** position $D$ and **tips donated** $T$ per each customer was in his list. Then, he asks you what is the **maximum** profit he earned today.

**Note**

if Mori at position $x$ he can only pick up a passenger in position $p_i \geq x$

## Input

The input begins with the integer $t$, the number of test cases.

The first line of each testcase contains a single integer $N$ where $(1 \leq N \leq 10^5)$. Then follows $N$ lines that will contain 3 values per each $P_i$, $D_i$ and $T_i$ where $(1 \leq P_i < D_i \leq 10^9)$ and $(1 \leq T_i \leq 10^9)$.

## Output

For each test case, Print a single line contains the **maximum** profit **Mori** can earn that day.

## Example

| standard input | standard output |
|---|---|
| 1<br>6<br>0 5 1<br>2 9 2<br>9 11 3<br>10 11 2<br>11 14 2<br>12 17 1 | 20 |

## Note

First example:

The way to earn the most money is by accepting passengers at indices 1, 2 and 5.

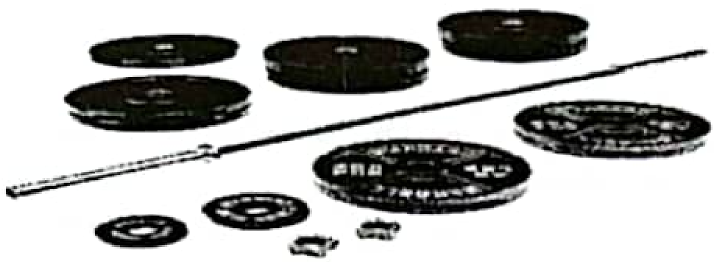
- The amount paid by the passenger at index 1: 9 - 2 + 2 = 9
- The amount paid by the passenger at index 2: 11 - 9 + 3 = 5
- The amount paid bv the passenger at index 5: 17 - 12 + 1 = 6
- The total amount paid by the passengers is 9 + 5 + 6 = 20

Therefore, the return value is 20.

# Problem G. lifting weights

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Balloon Color: | `Gold` |

We are opening a GYM center for athletes soon. Now, we have bought **N barbells (weights)** in our GYM to be used for the bars. As you may know, to use the bar you need to have **equal** weights on each side.



We already have a barbell of weight **X** on the first side.

Now we need to find a group of barbells that can be placed on the other side such that it has a maximum capacity that doesn't exceed **X**.

Can you help us to get the **maximum capacity** that meets the requirement above?

## Input

The input begins with the integer $t$, the number of test cases. Then $t$ test cases follow

The first line of each testcase will contain two numbers $N$ and $X$ where $(1 \le N \le 35, 1 \le X \le 10^{18})$.

The second line will contain $N$ numbers where $(1 \le N_i \le 10^{18})$

## Output

For each test case, Print only one line containing the maximum capacity that meets the requirements above.

## Examples

| standard input | standard output |
|---|---|
| 1<br>5 7<br>7 1 5 6 2 | 7 |
| 1<br>4 11<br>7 1 5 2 | 10 |

## Note

First example:

There are 3 ways to reach the maximum weight: 7, 1, 6 and 2, 5. Other combinations are either more or less than 7 combined

# Problem H. Thank you, sir

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Balloon Color: | `White` |

There are $n$ students in her microbiology class. The $ith$ student will say "Thank you, sir!"only after at least $Ai$ other students have also said, "Thank you, sir!"

if $Ai$=0, then the $i_{th}$ student says "Thank you, sir!"immediately after class ends.

Edward calls this behavior decorum sensing because these students are trying to get a feel for what is appropriate to say or do based on what everyone else is doing.

## Input

The input begins with the integer $t$, the number of test cases. Then $t$ test cases follow

The first line of each testcase will contain only one integer $N$ where ($0 \le N \le 10^5$).

The second line will contain $N$ integers where ($0 \le A_i \le 10^9$).

## Output

For each test case, find the number of students who will eventually say "Thank you, sir!"

## Example

| standard input | standard output |
|---|---|
| 1<br>5<br>0 1 2 3 4 | 5 |

# Problem F. Minimum swaps

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Balloon Color: | `Yellow` |

Programming competition are very important these days. It helps you to improve your thinking way and your problem solving skill. Therefore, we decided to make our own string competition. We need you to construct the most **beautiful** string using the given **binary** string $S$ of size $N * N$ where $N$ is an even **integer** and $S$ contains an equal numbers of 0s and 1s.

# 10010110

In order to do such, you are allowed to swap any two digits $S_i$ and $S_j$ where $(0 \leq i, j < N * N, i \neq j)$ such that the final string meets the following:

- If $((i+1) \bmod N)$ - $(i \bmod N) = 1$, then $(S_{i+1} + S_i) = 1$ for $(0 < i < (N * N) - 1)$
- If $(i + N) < (N * N)$, then $(S_{i+N} + S_i) = 1$.

**Note:** Assume the string is 0 based indexing.

Could you construct this **beautiful** string with the **minimum** number of swaps required to do such an operation.

## Input

The input begins with the integer $t$, the number of test cases. Then $t$ testcases follow

The first line of each testcase will contain a single integer $N$ where $(1 \leq N \leq 10^3)$.

The second line will contain a string $S$ of size $(N * N)$ and consists of 0s and 1s.

## Output

For each test case, Print a single line containing the **minimum** number of swaps required to construct the **beautiful** string.

## Examples

| standard input | standard output |
|---|---|
| 1<br>2<br>0011 | 1 |
| 1<br>4<br>1010101010101010 | 4 |

# Problem I. Primes Every Where Hard Version

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Silver`

Given an array $a$ of size $n$.

Print the number of pairs $(i, j)$ such that $(a_i + a_j)$ is prime and $i \neq j$.

**Prime numbers are positive integers that have exactly two distinct positive divisors.**

## Input

The input begins with the integer $t$, the number of test cases.

The first line of each test case contains an integer $n$ denoting the length of the array.

The second line contains $n$ integers $a_1, a_2, \ldots, a_n$.

- $1 \le n \le 10^6$
- $1 \le a_i \le 10^6$
- $a_i$ is always prime.

## Output

For each test case, Print the number of pairs.

## Example

| standard input | standard output |
|---|---|
| 1<br>5<br>2 3 5 7 2 | 8 |

## Note

The first test : (1,2),(2,1),(1,3),(3,1),(2,5),(5,2),(3,5),(5,3).

# Problem J. Flip Grid

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Balloon Color: | `Orange` |

There are $R * C$ coins are arranged into a grid with $R$ rows and $C$ columns. The coins are numbered in row major order, starting from zero. (That is, top row from left to right has coins from 0 to $C - 1$, second row has coins $C$ to $2C - 1$, and so on).

Initially, Coin 0 is heads up and all other coins are tails up.

You want to remove all coins, one after another. There are two rules:

- You can only remove a coin if it's heads up.
- Whenever you remove a coin, you must flip all coins that are **horizontally or vertically adjacent**.

Print an array that contains the numbers of coins in an order in which they can be removed. Any valid solution will be accepted. If there is no solution, print "**Impossible**".

## Input

The input begins with the integer $t$, the number of test cases. Then $t$ test cases follow.

The only line of each test case contains two integers $R$ and $C$ where $(1 \leq R, C \leq 30)$

## Output

For each test case, Print two lines. The first containing the array size. The second containing the array required above. If there is no such array print "**Impossible**".

Any valid solution will be accepted.

## Example

| standard input | standard output |
|---|---|
| 1<br>3 3 | 9<br>0 3 4 6 5 2 8 7 1 |

## Note

The nine coins are numbered as follows:

012

345

678

Below we illustrate the example return value by showing the state of the board at the beginning and also after each step. ('H' denotes a coin that's heads, 'T' denotes a coin that's tails, and '.' a coin that has already been removed.)

```
0:   1:   2:   3:   4:   5:   6:   7:   8:   9:
HTT  .HT  .HT  .TT  .TT  .TH  .H.  .H.  .H.  ...
TTT  HTT  .HT  ..H  ..H  ...  ...  ...  ...  ...
TTT  TTT  HTT  HHT  .TT  .TH  .TH  .H.  ...  ...
```

There are other valid solutions, too.

# Problem K. Check Names

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Dark Blue`

Our precious coaches '**ezzat**', '**hussien**', '**hossam**' and '**omar**' decided to take responsibility of making a competitive charm contest. They are going to give you a string $S$ that consists of **lowercase letter** and **dots(.)**.

Your goal is to determine if there is any possible way to replace all the **dots(.)** if it exists in the string with any lowercase letter to construct one of given coaches names above. If such goal can be achieved print the **coach name** otherwise, print **Impossible**.

## Input

The input begins with the integer $t$, the number of test cases. Then $t$ test cases follow.

Each test contains a single line with a String $S$.

## Output

For each test case, Print a single line contains the answer required above.

## Examples

| standard input | standard output |
|---|---|
| 1<br>ah.ed | Impossible |
| 1<br>hussien | hussien |

# Problem L. Maximum Work

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Balloon Color: | `Bronze` |

Bob has $N$ workers working under him. The $i^{th}$ worker has some strength $W_i$.

Currently, he has $M$ tasks pending with him. Each task can be assigned to only 1 worker.

To do a particular task $i$, Strength $S_i$ is required i.e. a task $i$ can be done by a worker only if his strength is greater or equal to $S_i$.

Bob has $K$ magical pills, taking a magical pill will increase the strength of a worker by $B$ units.

Bob gives pills to workers optimally such that the maximum number of tasks can be done.

Determine the maximum number of tasks that can be completed.

## Input

The input begins with the integer $t$, the number of test cases.

The first line of each testcase line will contain four integers $N$, $M$, $K$ and $B$ where $(1 \leq N, M, K, B \leq 10^5$

The second line will contain $N$ integers where $(1 \leq W_i \leq 10^5)$, where $W_i$ represent the strength of the $i$ worker.

The third line will contain $M$ integers where $(1 \leq S_i \leq 10^5)$, where $S_i$ represent the strength required the $i^{th}$ task.

## Output

For each test case, Print a single contains the answer required above.

## Examples

| standard input | standard output |
|---|---|
| `1`<br>`2 2 2 1`<br>`3 2`<br>`2 3` | `2` |
| `1`<br>`3 3 10 2`<br>`10 5 1`<br>`5 19 5` | `3` |

# Problem M. Invert Grid

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Balloon Color: | `Pink` |

We are developing a new game that helps our children to improve their thinking skills. The game rules are as follows:

- You have a grid of size $N * M$ that consists of $0s$ and $1s$.
- The grid is 0 based.
- In each move you can choose a grid cell $G_{R,C}$ then, invert all the cells $G_{i,j}$ where $(0 \leq i \leq R, 0 \leq j \leq C)$.
- The final grid cells should contain 0 values after applying any number of moves.

Note: Inverting a value means changing the 0 to 1 and vise versa.

Sometimes, the game that is generated by our software is hard. Your child asks you to solve this game to take the lead of the score board. Score is going to be high if you manage to solve this problem with minimal number of moves.

## Input

First line will contain two integers $N$ and $M$ where $(1 \leq N, M \leq 50)$.

Next $N$ lines will contain $M$ digits consist of $0s$ and $1s$.

## Output

Print the minimal number of moves you have to do in order to solve the game.

## Examples

| standard input | standard output |
|---|---|
| 2 4<br>0000<br>0000 | 0 |
| 2 2<br>01<br>01 | 2 |

3 3 10
10

0,1
0
0,0
R=0, C=0

# Problem N. Jumps Again

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Balloon Color: | `Light Green` |

You are given a rectangular board where each cell contains either an integer between 1 and 9, inclusive, or a hole.

Place a token into the cell in the upper left corner of the board. Now you can play a simple game. The game consists of moves, and each move looks as follows:

- Look at the number X written in the cell where your token is placed.
- Choose one of the four basic directions (up, down, left, or right).
- Move your token exactly X cells in the chosen direction. You can jump over all intermediate holes in the path.

The game ends after a move that lands the token in a hole or outside the board. Your goal is to make as many moves as possible.

The board is given as a rectangular grid of characters. Characters '1' to '9' represent cells containing the corresponding integer, and letters 'H' represent holes. The upper left corner of the board corresponds to the first character of the first element of board.

Output the maximum number of moves you can make on the given board. If it is possible to make a arbitrarily large number of moves, output -1.

## Input

The first line of the input contains an integer $T$ the number of test cases($1 \leq T \leq 10$).

Next $T$ lines contains $N, M$ the size of the board ($1 \leq N, M \leq 50$). following by $N$ lines each contains characters.

each cell on the board either an integer between 1 and 9 or $H$, where $H$ reprsents a hole.

## Output

Output $T$ lines the answer required above.

## Example

| standard input | standard output |
|---|---|
| 1<br>2 6<br>14HHH9<br>2H3HH1 | 5 |

1111
1211

E, F, G, M, O

# Problem O. Shared Interest

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Balloon Color: | `Light Purple` |

Given a group of friends who have different interests, determine which pair of friends have the most interests in common.

Then use a little math to determine a value to return.

The group will be represented as a graph with nodes numbered consecutively from 1 to $n$ that represents each member.

Friendships have evolved based on interests which will be represented as the weights of the edges in the graph.

Any pair of members who share the same interest are said to be connected by that interest. Once the node pairs with the maximum number of shared interests are determined, multiply the resulting node pairs and return the maximal product.

## Input

The first line contains an integer $T$ the number of test cases.

next $T$ line each contains:

a line contain two integers $N$ and $M$ where $(2 \leq N \leq 10^5, 1 \leq M \leq 10^5)$.

followed by $M$ lines will contain three integers $u_i$, $v_i$ and $w_i$ where $(1 \leq u_i \neq v_i \leq N, 1 \leq w_i \leq 10^5)$. represent that friend $u_i$ share interest number $w_i$ with friend $v_i$.

## Output

Print $T$ lines the answer required above.

## Example

| standard input | standard output |
|---|---|
| 1<br>4 5<br>1 2 2<br>1 2 3<br>2 3 1<br>2 3 3<br>2 4 5 | 6 |

## Note

2 : 1
3 : 2
1 : 1
5 : 1

In the example,

nodes $(1, 2)$ share two different interests $(2, 3)$

nodes $(2, 3)$ share two different interests $(1, 3)$

nodes $(2, 4)$ share 1 interest $(5)$

Thus nodes $(1, 2)$ and nodes $(2, 3)$ have the most interests in common, so we return $2 * 3 = 6$ which is the maximum product