CODING NOW
PROGRAMMING FUTURE

6th Jilin Province Collegiate Programming Contest

第六届吉林省大学生程序设计竞赛

May 20, 2012

1	Problem A	Welcome 2012
2	Problem B	Chinese Knight
3	Problem C	The Largest Arrow
4	Problem D	Nurse Scheduling
5	Problem E	Teleportation
6	Problem F	Shadows
7	Problem G	RP Path
8	Problem H	Plants vs. Zombies
9	Problem I	Crossword
10	Problem J	Magic Sequence

Notice: If the problems have been patched, please refer to the web page
注：如有修改，以网上实时页面为准。

Problem A: Welcome 2012

Welcome to 2012 Jilin Province Collegiate Programming Contest.

Year 2012 is a special year, and the 2012 is also a special number. For example, 2012= 2*2*503, here 2 and 503 are both prime number.

As we know, A prime number (or a prime) is a natural number greater than 1 that has no positive divisors other than 1 and itself.

This time, we should find out the numbers that can be divided like 2012. That is to say, for a positive number n, can be written *n=p1*p1*p2*, here *p1* and *p2* are different prime numbers.

Input

This problem has no input

Output

You should output all the numbers that satisfied the rules before. There numbers are increasing order and less than 3000.

Sample Output

```
12
18
......
2012
......
```

Not all the numbers are listed in the sample. There are just some examples. The ellipsis expresses what you should calculate.

Problem B: Chinese Knight

There are different pieces In Chinese Chess, such as General, Advisor, Elephant, Knight, Chariot, Cannon and Solider.

The knights(horses) are labeled 馬. They begin the game next to the elephants. A knight moves and captures one point orthogonally and then one point diagonally away from its former position, a move which is traditionally described as being like the Chinese character '日'. The knight does not jump as the knight does in Western chess. Thus, if there were a piece lying on a point one point away horizontally or vertically from the horse, then the horse's path of movement is blocked and it is unable to move in that direction. Note, however, that a piece two points orthogonally or a single point away diagonally would not impede the movement of the horse. Blocking a horse is also known as "hobbling the horse's leg". The figure 2.1 illustrates the horse's movement.

Since knight can be blocked, it is sometimes possible to trap the opponent's horse. It is possible for one player's knight to attack the opponent's horse while the opponent's knight is blocked from attacking, as seen in the figure 2.2.

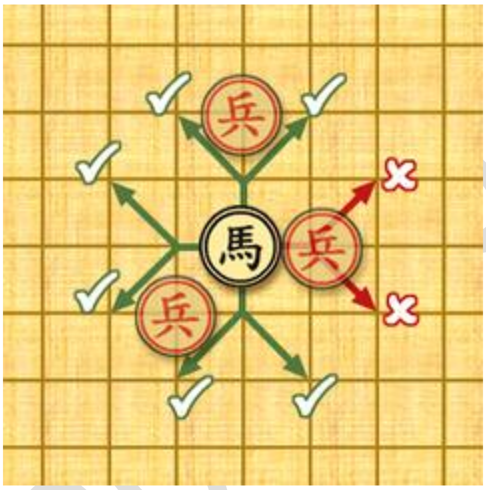

Figure 2.1 Movement of knight

Figure 2.2 Asymmetric movement

On a n*n grid chessboard, a Chinese knight want to visit each placement exactly once. This is just like traditional "Knight's tour problem". Unfortunately, there are some obstacles on the placement. The knight can't reach it's placement and maybe blocked by these obstacles. On the other hand, it is also fortunate, the number of all possible paths is decreased rapidly.

Our knight will always begin his tour at top-left. The number of obstacles is between 1 and 5. You should calculate the number of valid paths. It is not necessary that the knight return his starting position. And all the obstacles is excluded from the path.

Input

The first line of each case are three integers *m, n* and *k*. *m* and *n* are the size of chessboard (*m, n<=6*) and k is indicated the number of obstacles. k is -1 means the end of input.

The next k lines are two numbers that are coordinates of each obstacle (starting from 1).

Output

For each case, output the number of all successful paths.

Sample Input

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

Sample Output

```
2
```


Hint

One of possible path of the sample input is described as follows. Each number is the step of knight except -1 means obstacle.

1	18	13	6	23
12	7	-1	19	14
17	2	9	22	5
8	11	4	15	20
3	16	21	10	-1

Problem C: The Largest Arrow

Three points $A(x_1, y_1), B(x_2, y_2)$ and $C(x_3, y_3)$ form an arrow if and only if the following conditions are fulfilled:

$x_1 = x_3 < x_2$

$y_1 < y_2 < y_3$

The length of the arrow $(x_2 - x_1)$ must not shorter than the width of the arrow $y_3 - y_1$

Figure 3.1 shows an example of an arrow. We define the size of an arrow as its length plus its width. Given N points in the plane, what's the largest possible arrow can be formed from three of the given points?

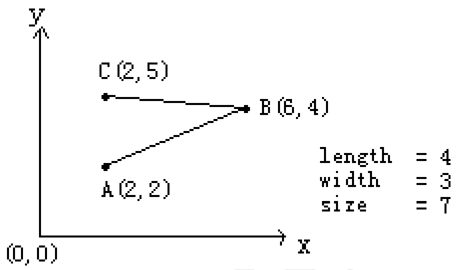

Figure 3.1: Example of an arrow

Input:

The first line of the input contains a positive integer T specifying the number of test cases to follow.

The first line of each test case contains a positive integer N ($1 \leq N \leq 2000$). Then follows N lines, each describes a point with coordinates (x_i, y_i). ($0 \leq x_i, y_i \leq 10^7$)

It's assumed that no two points are at the same location.

Output:

For each test case you should output a single line containing "Case X: Y " (quotes for clarity) where X is the number of the test case (starting at 1). Y is the size of the largest arrow in the plane. If there's not any arrow in the plane, Y should be equal to -1.

Sample Input:

```
1
5
2 2
```

```
2 5
6 4
2 4
2 10
```

Sample Output

```
Case 1: 7
```

Problem D: Nurse Scheduling

There are N nurses to be scheduled for 5 days. To be precisely, each nurse will be assigned a working pattern, which is a 0/1 string of length 5 indicating which days she will work on. For example, if a nurse is assigned a pattern denoted as 01001, that means she will work on the second day and the fifth day, and will be off on the other days. Nurses have their own preferences about these working patterns. The preference, of a nurse on a pattern, will be denoted as a non-negative integer; the larger it is, the more that nurse like that pattern. Each of the 5 days has a certain requirement for nurses.

The task is to assigned every nurse a working pattern, meeting exactly the requirement of every day (no more or less), and make the sum of the preferences of all nurses on their corresponding assigned pattern as large as possible.

Note that, there are always *25=32* candidate working patterns for each nurse, that is, *00000, 00001, …, 11110, 11111*.

Input

First is an integer T, indicating the number of test cases. $T < 25$.

Each test case is formatted as following:

First is an integer N, indicating the number of nurses. $N < 21$.

N lines follow. The i-th line contains 32 non-negative integers, separated by space, denoting the preferences of the i-th nurse on the patterns of *00000, 00001, 00010, …, 11110, 11111*, respectively. All preferences are less than *70000*.

The last line of a test case contains 5 non-negative integers, indicating the requirements for the 1st, 2nd, 3rd, 4th, 5th day, respectively. It's guaranteed that there always exists a feasible solution.

Output

For each test case, print one line containing the largest sum of preferences.

Sample Input

```
1
1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
0 0 0 1 0
```

Sample Output

```
3
```

Problem E: Teleportation

After defeating Freeza, Son Goku managed to survive from the explosion of the planet Namek. He landed on the planet Yardrat. He found the people there have an interesting ability: teleportation, which can transmit one from a place to another instantly. He asked someone to teach him this ability. Now they start practicing.

There is a tall building, which can be considered as an $n \times m$ grid. The grid is labeled $0 \sim n-1$ from top to bottom and $0 \sim m-1$ from left to right. They start from the top of the building. The Yardrat teacher stands on the left and Goku on the right. Each time they teleport to some column of the next row below them. As Goku is a novice, there are some constraints. Firstly, the teacher should be left to Goku. Secondly, the distance between the teacher and Goku can neither be less than mi nor be more than ma. That is to say, if the teacher is at the column i, Goku must be at the column in the range of $[i+mi, i+ma]$. Thirdly, not all the columns in the row is teleport-able. Luckily, they know which column can be teleport to in each row.

Each teleportation needs energy. When teleportation from column i to column j in the next row, the cost of energy is |i - j| (absolute value of $i-j$). Goku wants to cost the least energy, but he is no so clever to plan the route of teleportation. So he turns to you. Please help Goku minimize the total energy cost of him and the teacher to teleport from row 0 to row n - 1.

Input

There are multiple input cases. The first line is a positive integer, the number of test cases.

In each test case, the first line are 6 positive integers n, m, mi, ma, x, y $(1 \le n \le 1000,\ 3 \le m \le 10,\ 1 \le mi \le ma < m,\ 0 \le x < y < m)$. x is the initial position of teacher and y is that of Goku.

Following are n lines, each line has m characters. The characters can be `'*'` or `'.'`. `'*'` means Goku and the teacher can transport to the position while `'.'` means they can't.

Output

For each test case, output an integer in a line. If the teacher an Goku can reach the bottom, output the least total energy cost of them, otherwise output -1.

Sample Input

```
2
6 4 1 2 0 2
*.**
..**
.*.*
****
***.
**.*

5 5 1 3 0 4
*****
*..*.
***.*
*.*.*
*....
```

Sample Output

```
6
-1
```

Problem F: Shadows

"No sunshine but bath some shadow". A shadow is an area where direct light from a light source cannot reach due to obstruction by an object. It occupies all of the space behind an opaque object with light in front of it.

Let us consider a simplified 2-D problems. There is a lamp at the origin and a unlimited wall on the straight line x=d. Some opaque objects float between them. All these objects are circle. The wall will be separated into light and shadow. The shadow of each circle may be overlapping no matter the circles are intersected or not.

Your task is to count the length of all shadows.

Input

The first line of each case are d and n. The real number d is the distance between wall and lamp, and integer n $(0<n<100)$ is the number of circles. n=0 means the end of input.

The next n lines consist of three real number x, y *and* r. x and y are the coordinates of circle center. r is the radius of circle opaque object. The entire area of each circle is between the two straight lines x=0 and x=d;

Output

For each case, print the total length of all shadows, rounded to two digits after decimal point.

Sample Input

```
10.0 2
5 0 2
3 1 2
10 0
```

Sample Output

```
20.16
```

Problem G: RP Path

There are N islands connected by N-1 bridges. And for every two islands there's one and only one unique path(NOT edge) to travel between them through bridges. It's not difficult to figure out that there are $N*(N-1)/2$ paths in total.

Some of the bridges are built a long time ago and it is dangerous to pass any of them. Those bridges are called RP (Repair-Prepare) bridges, which is also means that you should need RP to pass them alive.

If the path connecting two islands includes at least one RP-bridges, this path is called a RP-path.

Now you want to minimize the number of RP-paths by repaired one bridge at most. What is the minimal number of RP-paths you can archive ?

Input

The input contains multiple cases.

The first line of each input is a positive integer N $(1 \le N \le 200000)$, which is the number of islands.

There are N-1 lines followed. Each line containing 3 integers A,B and C, indicating a bridge connecting island A and island B. If this bridge is a RP-bridge C=1, otherwise C=0.

Output

For each case, output the number of minimal number of RP-paths in one line by repaired one bridge at most.

Sample Input

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

Sample Output

```
7
```

Problem H: Plants vs. Zombies

We all love playing the famous game: Plants vs. Zombies. Hovey is wandering how to solve a simplified version of this game.

The game is set in a chess board with N rows and M columns. Both the plants and zombies are located in some of the cells. In this version, all the actions are considered by rounds. One round contains two phrases, the plant's attack phrase following by the zombie's move phrase.

In this version, there is only one kind of plant. The attack range of them is described by a *(2K+1)*(2K+1)* square, whose center is the plant's position.(see Table 8.1)

Attackable	Attackable	Attackable
Attackable	Plant, Attackable	Attackable
Attackable	Attackable	Attackable

Table 8.1. Attack range with K = 1

When the zombies get into any one of the range of plants, they will get hurt by ATT points of HP in each round, no matter how many plants can attack them at the same time.

Luckily, the plants are invisible, so they won't be eaten by the zombies. However, they cost different sun points(C_{ij}) to be planted in different position. If the cost is zero, it means that you can't plant in the corresponding position. There is at most one plant for each position as well. Here come your lovely zombies. There is only one kind of zombie. They own different HP, but walks in the same way: move to the left adjacent cell in the same row in each round. When they move out of the board, the game is over.

Before the zombies' invasion, you will be given finite SP sun points to plant, and your task is to prevent the zombies from passing across from right to left.

During the zombies' invasion, the zombies will appear one by one at the right most cell of the board. i.e. The next zombie will not appear until the current zombie get eliminated(HP<= 0).

Given the order of the zombies, along with their appearing row and HP, you are going to give a best solution of planting, with given SP sun points, to eliminate maximum zombies, before the zombies' invasion. Also, you need to output the minimum cost if there're multiple solutions.

Input

There are multiple test cases. The first line contains the case number T.

In each case, in the first line, there are five integer indicating the number of rows

$N<=100$, the number of column $M<=100$, the attack range $0<=K<=3$, the initial sun points $1<=SP<=10000$, the damage $ATT<=10000$.

There follow N row, with M integers each line, indicating the cost $0<=C_{ij}<=10000$ to plant in row i and column j. It's guaranteed that at most two positive integers in each row.

A single number $1<=R<=1000$ follows in a line, indicating the number of zombies in the invasion.

R lines follow, each line contains two integers, the appearing row $0<=R_i<N$, and the total $1<=HP_i<=10000$. All the zombies appear in the same order as input.

Output

There are T lines in the output.

Each line contains two numbers separated by one space, the maximum number of zombies can eliminate, and the minimum total cost of planting among the best solutions.

Sample Input

```
1
4 5 1 15 5
0 0 4 0 0
0 5 0 7 0
0 0 0 0 0
0 8 0 0 6
5
0 20
1 20
2 25
3 10
0 25
```

Sample Output

```
4 15
```

Problem I: Crossword

Xiao Ming is very busy. Xiao Fang, one of his classmates, want to send some small paper pieces to Xiao Ming. She doesn't want to let other students know these private messages. So, she encrypt the message using the pattern followed. She draws a minimum $n*n$ square that can contain total characters at first. Secondly she writes these characters from left to right, top to bottom. At last, she rewrite these characters in square to a new string from top to bottom, left to right. The new string is passed to Xiao Ming.

Let us see an example:

If Xiao Fang want to write "ABCDEFG", she draw a 3*3 (3 is most suitable) square, and fill letters in it as follows:

ABC

DEF

G

At last, she rewrite these letters as "ADGBECF". All the blanks are omitted.

Can you write a program to help Xiao Fang encrypt her private message?

Input

The first line is the number of test cases. Each line followed is a string which only consist alphabet ('a'-'z', 'A'-'Z'). The length of string is no more than 100.

Output

For each case, print the encrypted string in one line.

Sample Input

```
2
ABCDEFG
abcdefghijk
```

Sample Output

```
ADGBECF
aeibfjcgkdh
```

Problem J: Magic Sequence

Ivankevin loves sequences.

A sequence of n + m elements is called a MAGIC SEQUENCE if:

1. Each element $a_i (1 \leqslant i \leqslant n+m)$ is +1 or -1.
2. There are total n “+1”s and m “-1”s.
3. For every $j (1 \leqslant j \leqslant n+m)$, we have the sum of $a_i (1 \leqslant i \leqslant j)$ is nonnegative.

Now given n and m, Ivankevin wants to know the number of magic sequences consists of n “+1”s and m ”-1”s.

Input

The first line is an integer T, the number of test cases.

Each case is a line of two integers n and m, 1≤n,m≤100.

Output

For each case, print an integer x, which is the number of magic sequences modulo 100000007.

Sample Input

```
2
2 1
3 3
```

Sample Output

```
2
5
```