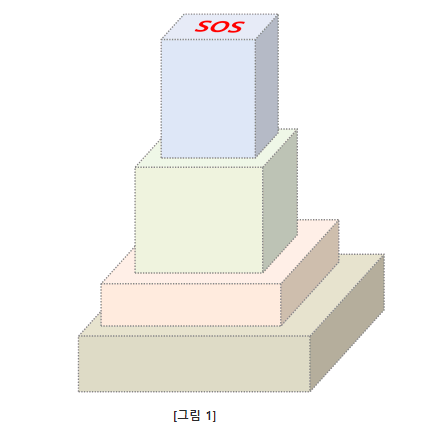
I was traveling on a boat and I met a typhoon.

The ship was wrecked, and I could have managed to escape by using a lifeboat,

At the end of drifting, it was brought to the uninhabited island.

Namsan, who is trapped on the uninhabited island, wants to send a SOS signal to a plane passing by nearby sea.



When you escape the lifeboat, you got N cubes

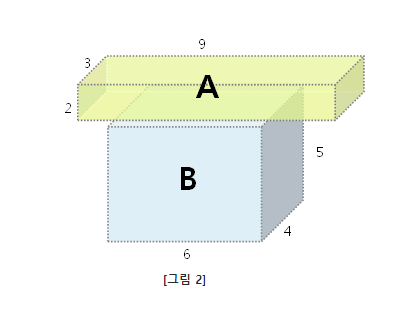
As shown in [Figure 1], I try to make it look like a tower by stacking it like a tower.

The rules for stacking boxes are as follows.

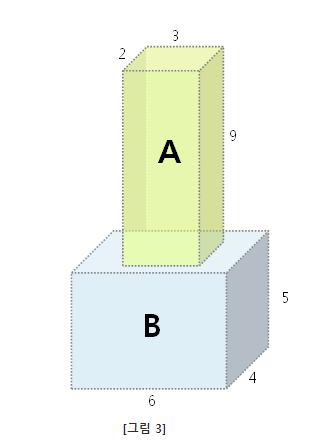
1. Each box can be stacked by rotating it by 90 degrees with respect to the horizontal, vertical, and height axes.

2. There are no restrictions on the stacking order of N boxes, and you do not have to use all the boxes.

3. The bottom of all boxes should not be taken away from the top of the stacked boxes.



For example, in [Figure 2], the bottom of box A is out of the top of box B, so it can not be stacked.

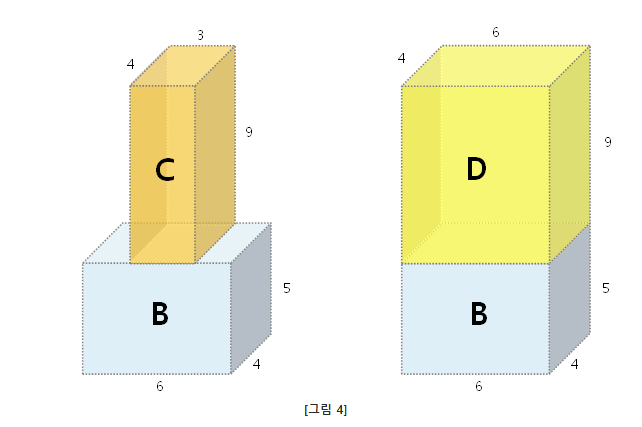


However, if the box A is rotated so that the narrow side faces downward as shown in [Figure 3]

Since the bottom surface of box A does not deviate from the top surface of box B,

In this case, box A can be stacked on box B.

[Figure 4] shows the case where the length of one side or two sides of the box placed above and the box placed below match



The bottom of box C is equal in length to the top of box B.

The bottom of box D and the top of box B have the same length on both sides.

Even in such a case, since the bottom surfaces of the boxes C and D placed above do not deviate from the top surface of the box B, the box can be stacked.

===

Given the lengths of the N boxes, width, height, and height, respectively,

Write a program that calculates the maximum height that can be stacked.

[Constraint]

1. The number of boxes N is an integer from 2 to 20 inclusive. (2 & amp; le; N & amp; le; 20)

2. The length of one side of the box is 1 or more and 10,000 or less.

?

[input]

Input is the total number of test cases T in the first line.

Each test case is given from the next line.

The first line of each test case is given the number N of boxes,

From the second line, the length, height, and length of each box are listed.

[Print]

The result for test case t is "#t", one space, and the correct answer.

(where t is the number of the test case and starts at 1)

The maximum height that can be created by stacking the boxes is output as the correct answer.

OLD SMART PHONE

I want to enter the desired number on my smartphone.

However, the smartphone is outdated and no part of the screen is touchable.

So, you may not be able to enter the desired number directly.

However, in such cases, there is no method.

This is because you can enter the numbers you can input in the calculator app and perform the calculations to create the desired number.

You can copy and paste it.

For example, as shown in <Figure 1>, '5', '6', '8', and '9'

You can input only '0', '1', '2', '3', '4', '7', '+', '-', '\*', '/' Let's say.

If you want to create a number 5, touch '2', touch '+', touch '3' and touch '='.

Touch a total of 4 times.

Of course, you can make 5 in another way.

Touch '1', touch '+', touch '2', touch '+', touch '2' and touch '='

However, in this case, touch six times. You have to touch more than the previous method.

In this example, the minimum number of touches to make 5 is 4.

You can not make 5 with a touch smaller than that.

If you can touch the number '5', you can make 5 by touching '5' once, in this case the minimum number of touches is 1.

**When you want to create a number in the calculator app** like this **, write a program to find the minimum number of touches** .

The calculation range of the calculator app is only 0 to 999.

**If the result of the input number or calculation is negative or exceeds 999, a problem occurs** .

If you want to enter more than two digits, you can touch the digits continuously.

If you want to make 41, you can enter '4' 1 '.

The operations are performed sequentially.

If '2', '+', '2', '\*', '3', '=' is entered,

The division divides the fraction less than the decimal part and leaves only the integer part.

For example, '7', '/', '3', '='

If '1', '/', '5', '=', the result is 0.

Also, it should not be divided by 0.

Touchable numbers consist of numbers from 0 to 9. Touchable operators consist of operators '+', '-', '\*', '/'.

'=' Is always touchable and input is possible.

You must enter '=' to get the result of the operation.

**You do not have to type '=' if you can enter numbers directly to create the number you want** .

The initial screen of the calculator app does not display any numbers.

Also, the calculator app has an input limit, and the maximum number of touchable times is up to M.

**Touch the number is not to exceed M** .

If the desired number W can not be created until the touch count reaches M, -1 is output.

**[Restrictions]**

1. The calculation range of the calculator app is 0 to 999. The result of the operation during a number or calculation must not be negative or exceed 999.

(For example, if "999 + 2 - 3" is input, the calculation result becomes 1001 and the calculation range is exceeded.

2. The number of touchable numbers N is given as a natural number from 1 to 10 inclusive.

3. The number O of touchable operators is given as a natural number from 1 to 4.

4. The maximum number M of touches is given as a natural number between 4 and 20 inclusive.

5. Touchable numbers have integer values ​​between 0 and 9.

6. Touchable operators are represented by '+' for 1, '-' for 2, '\*' for 3, and '/' for 4.

7. '=' is always touchable.

8. Division The '/' operation takes only integer parts and should not be divided by zero.

9. The desired number W is an integer from 0 to 999.

10. You do not have to type '=' if you can enter numbers directly to create the desired number.

(For example, if '1' and '9' are inputable and the desired number is 91, you can make 91 by touching '9', '1' twice.)

**[input]**

The first line of input is the total number of test cases T (1 ≤ T ≤ 50).

From the next line, there are T test cases. Each test case consists of 4 lines.

The first line is the number N of touchable numbers, O the number of touchable operators, and M the maximum number of touches possible. There is a space between the N, O, and M values.

The next line comes with N touchable numbers. Touchable numbers are separated by spaces.

The third line comes up with O touchable operators. Touchable operators are also separated by spaces.

The fourth line is the desired number W.

**[Print]**

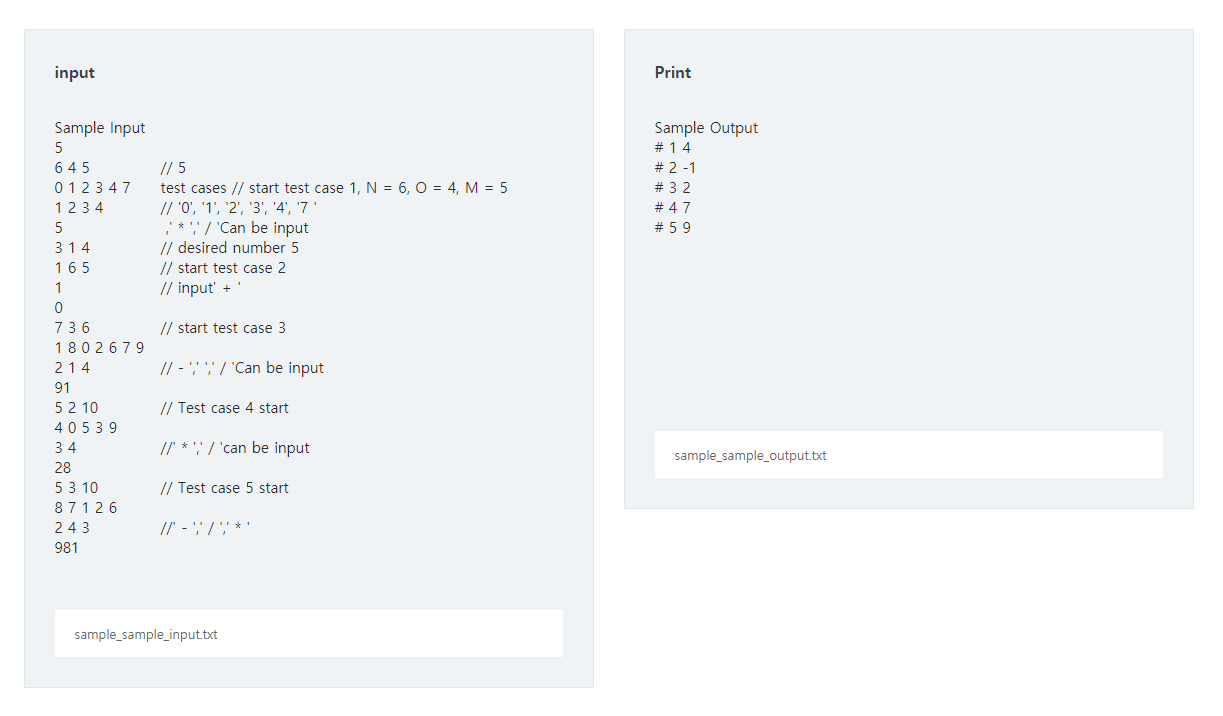
The result for test case T is "#T", one space, and the correct answer. (T stands for the number of the test case and starts from 1)

The correct answer is the minimum number of touches you can make the number W you want.

If the desired number W can not be generated within the maximum number of touchable times M, -1 is output.

The formula for the minimum number of touches for the Test Case of Sample Input is as follows.

|  |  |  |
| --- | --- | --- |
| Test Case | Desired number | Calculation formula at minimum touch |
| One | 5 | 1 + 4 = |
| 2 | 0 | Not possible. Can not be zero because the result of the operation is increasing |
| 3 | 91 | 91. You can create 91 by touching directly |
| 4 | 28 | 4 \* 35/5 = |
| 5 | 981 | 16 \* 62 - 11 = |



Samsung Electronics has discovered a new semiconductor material, and confirmed that chip performance is greatly improved when the semiconductor material is processed and manufactured into a chip.

A new semiconductor material is processed into a rectangular wafer and chip production is performed on the wafer.

Due to the introduction of a new wafer processing process, unusable regions will occur in the middle of the wafer, and the chip must be produced in the remaining regions.

**To produce one chip, a 2 \* 2 area is required on the wafer** .

  Assume that wafer information is given as in [Figure-1] below. In [Figure-1], gray represents the unusable area on the chip.

  The maximum number of chips that can be produced on the wafer in [ Figure-1] is 9, and there can be several ways to select the area that can be used as a chip.

  One of several methods can be used for chip production as shown in [Figure -2.1] or [Figure -2.2].

  As another example, when the wafer information is given as shown in [Figure 3.1], the maximum number of chips that can be produced is 10, and it can be produced as one of several methods as shown in [Figure 3.2].

Write **a program** that **outputs the maximum** length (H) and width (W) of the wafer and the **maximum number of chips that** can be **produced** when wafer information is given .

**[ Restrictions]**

  1. Information on the wafer is given in the form of a rectangle (including square).

  2. The vertical length H is 5 or more and 10 or less. (5 & amp; le; H & amp; le; 10)

  3. The width W is 5 or more and 25 or less. (5 & amp; le; W & amp; le; 25)

  4. Chip-processable area is **given as** 0, and **unprocessable area is given as 1.**

**[ Input]**

  The first line of input gives the total number T of test cases.

  Each test case is given from the next line, and the first line of each test case is given a height H and a width W.

  The next H lines are given W wafer information for each row. 0 is a machinable region, and 1 is an unprocessable region.

**[ Output]**

The result for test case T is "#T", and **output the number of chips with a** single space and **maximum production.**(T stands for the number of the test case, starting at 1)



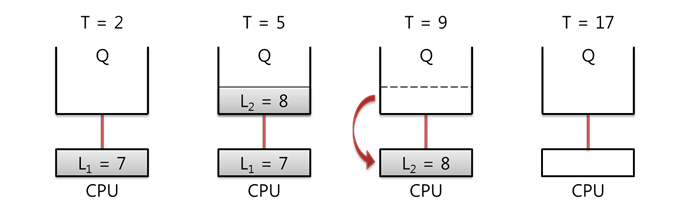
Samsung Electronics wants to upgrade its in-house network equipment.   
When the network equipment receives a packet, the CPU processes it , and the processing time is proportional to the length (L) of the packet .   
For example , the time required to process a packet of length 5 is 5 seconds .   
If CPU If it receives a new packet when no packet to complete the process , a packet queue (Queue) is stored in .   
After the CPU finishes processing the packet, it can process other packets stored in the queue . At this time, the waiting time in the queue is also included in the processing time of the packet .   
That is , the total processing time of one packet is the time spent in the queue plus the time processed by the CPU (T total = Tqueuing + T processing ).

You can use multiple CPUs to reduce latency in the queue . For example, assume that you received two packets as shown in the following table .

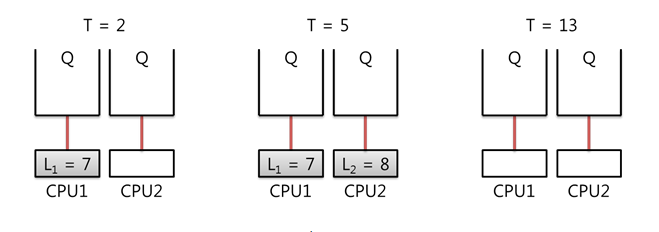
|  |  |  |
| --- | --- | --- |
| Packet number (i) | The reception time (T i ) | Packet length (L i ) |
| One | 2 seconds | 7 |
| 2 | 5 seconds | 8 |

[ Figure 1] and as CPU is 1 if individual , 5 at the beginning of 2 yet when it receives a time packet 1, since the times have not completed a packet processing status from the queue 4 is to wait for a second .   
Therefore, the total processing time of the second packet is 12 seconds including the waiting time . ( Waiting time 4 seconds + processing time 8 seconds = total 12 seconds )

                                                        [Figure 1]



But [ Figure 2] such as a CPU to 2 when used is one second immediately after receiving the packet time CPU2 because it can be processed by , the total processing time is equal to the packet length of 8 is the second .



                                                           [Figure 2]

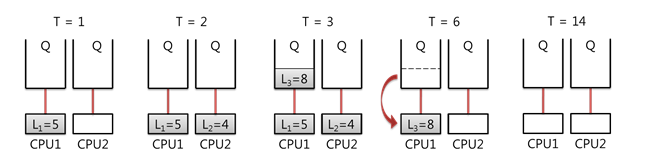
The longer packet processing time becomes poor when the work efficiency, the maximum processing time of each packet **10 or less seconds** , to establish a . Write a program to get the   
minimum number of CPUs needed .   
The maximum number of usable CPUs is 5 , and if all CPUs can not make processing time less than 10 seconds, -1 is output .

**[ Additional Example ]**

Multiple CPU If you use , each CPU has a separate queue .   
When a new packet arrives , allocating a packet to an empty CPU queue does not necessarily yield optimal results .   
For example, CPU the second packet according to the table below in the individual state 3 Let's assume that the received one .

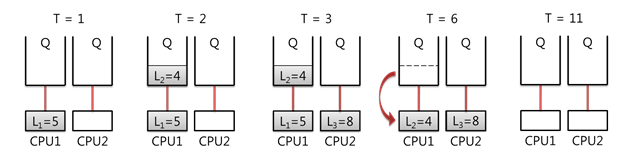
|  |  |  |
| --- | --- | --- |
| Packet number (i) | The reception time (T i ) | Packet length (L i ) |
| One | 1 second | 5 |
| 2 | 2 seconds | 4 |
| 3 | 3 seconds | 8 |

1, the packet times CPU1 while being processed in [ Figure 3] and as two times the packet CPU2 processed in the three times the packet CPU1 Suppose processed in .   
The total processing time of packet # 2 is 4 because there is no time to wait in queue. However , the total processing time of packet # 3 is 11 seconds including 3 seconds of waiting time .   
( Waiting time 3 seconds + processing time 8 seconds = 11 seconds )



                                                          [Figure 3]

However, [ Figure 4] and as two packets once CPU1 into a queue of the three packet times CPU2 when processed in two total processing time of the packet times the latency four total, including the second 8 man increase seconds ,   
three times a packet Can satisfy the requirement because the waiting time is lost and the process can be performed only in 8 seconds .



                                                           [Figure 4]

**[ Restrictions ]**

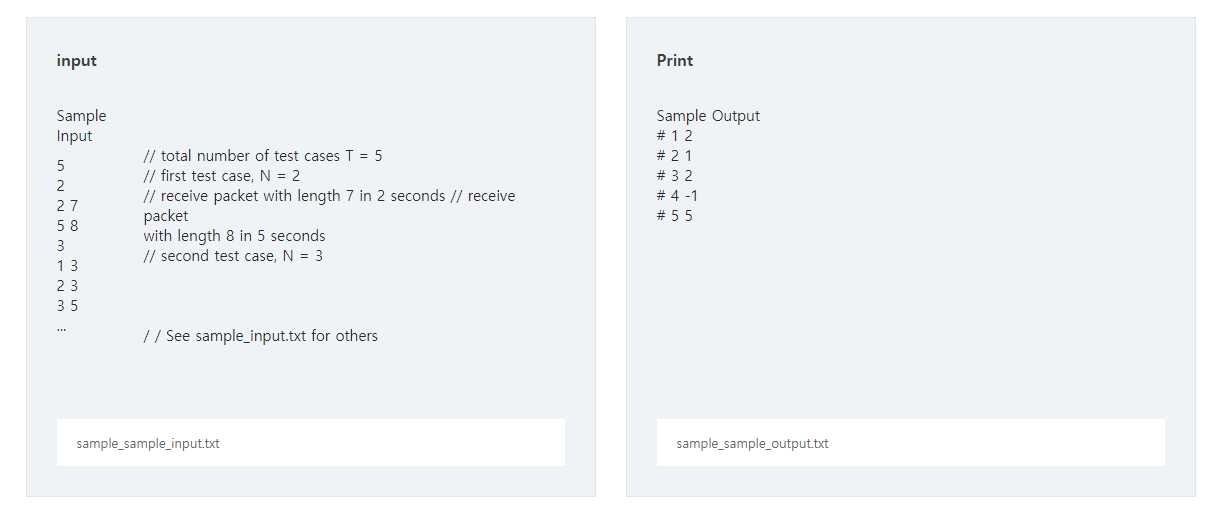
1. The number N of packets is an integer of 1 or more and 1000 or less . (1 ≤ N ≤ 100 0)   
2. The reception time T i of the i- th packet is an integer from 1 to 5000 . (1 ≤ T i ≤ 50 00) **( and a number of packets at the same time can be received, in which case is received in the device as the input sequence.)**  
    
3. i the length of the second packet L i is 1 more than 10 is an integer of not less . (1 ≤ L i ≤ 1 0)   
4. The maximum available CPU number is 5 is a dog .   
5.Each CPU has a separate queue .

**[ Input ]**

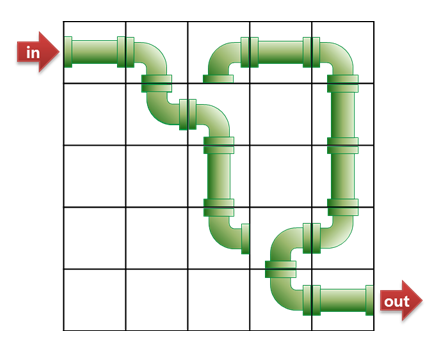
The first line of the input gives the total number of test cases T , and the next test case gives T test cases .   
The number of packets are the first line of a test case N is given , then the next line N in one line is received, given the time and length information of each packet .   
**The packet information is sorted in ascending order based on the reception time .**

**[ Output ]**

The output is "#t" , a space is displayed, and the correct answer is output . (where t is the number of the test case and starts at 1. ) The   
correct answer is the minimum number of CPUs required . If the maximum CPU number (5 gae ) with the processing time is also 10 if more than a second -1 and outputs .



[ Figure 1] as shown in the N \* N a pipe is arranged in the region .   
I want to rotate the pipe in each cell to connect the pipe from the top left to the bottom right .   
**Find a way to connect the pipe to the shortest length and write a program that prints the length of the pipe .**

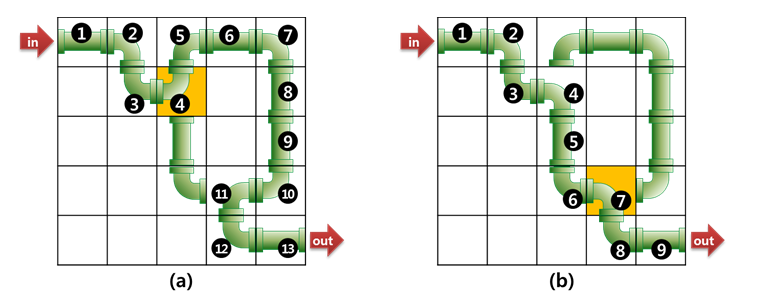


                                      [ Figure 1]

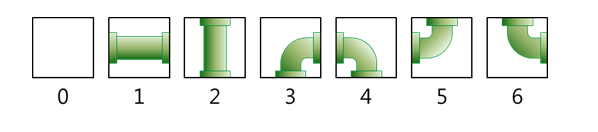
**[ Constraints ]**  
1. a side size of the area N is 5 or more 50 is an integer of not less . (5 ≤ N ≤ 50 )   
2. Straight pipe can be rotated 90 degrees , and bent pipe can be rotated 90, 180, 270 degrees .   
3. In the top left pipes ( starting points ) are necessarily left (in) to be connected to ,   
   the bottom right of the pipe ( the end point ) must be the right (out) to be connected and .   
4. Inputs that can not be connected between in and out are not given .

As shown in (a) of [ Figure 2] , if pipe of position ④ is rotated , it can be connected with length 13 . But this is not the optimal solution .

As shown in (b) of [ Figure 2] , if you pipe the pipe at position ⑦ , you can connect it with length 9 , which is the optimal method .



      [ Figure 2]

**The**  
first line of the **[ Input ]** input is given the total number of test cases T , followed by T test cases from the next line .   
The first line of the test case gives the size (N) of one side of the region .   
The pipe information is then given over the N lines .   
The number 0 is the cell with no pipe, and the other numbers are pipes with the following shape.  [ Figure 3]  
  


**[ Output ]**  
Outputs "#t" , displays a single space, and outputs the correct answer . (t stands for the number of the test case and starts from 1. ) The   
correct answer is the length of the pipe connected by the shortest length .

