Algorithms for Programming Contests
WS20/21 - Week 09

Chair for Foundations of Software Reliability and Theoretical Computer Science,
TU München

Mikhail Raskin, Martin Helfrich, Maxi Weininger, Christoph Welzel

Welcome to our practical course! This problem set is due by

Wednesday, 20.01.2021, 6:00 a.m.

Try to solve all the problems and submit them at

`https://judge.in.tum.de/conpra/`

This week's problems are:

A **Making Change** **1**

B **Bracelets** **3**

C **Packing Cases** **5**

D **Poker** **7**

E **Escaping the Paradox** **9**

The following amount of points will be awarded for solving the problems.

Problem	A	B	C	D	E
Difficulty	easy	easy	medium	medium	hard
Points	4	4	6	6	8

If the judge does not accept your solution but you are sure you solved it correctly, use the "request clarification" option. In your request, include:

- the name of the problem (by selecting it in the subject field)
- a verbose description of your approach to solve the problem
- the time you submitted the solution we should judge

We will check your submission and award you half the points if there is only a minor flaw in your code.

If you have any questions please ask by using the judge's clarification form.

This page is intentionally left blank.

Problem A
Making Change

Templonia is a very strange place. On one hand, there are beautiful temples, great ancient cities and many nice people. On the other hand, the Templonians have acquired some very peculiar customs. The one which causes the biggest problem for Lea is that Templonian merchants always want their money in as few coins or notes as possible. Even worse, there are many different coins and notes for the Templonian Column, and not every merchant takes all values. Surely, the Templonian people are accustomed to this behaviour and are very good in calculating these things. But as a foreign tourist, Lea has some difficulties to say the least. A tolerant person as she is, she does not want to anger the local merchants and tries to adopt Templonian customs. Whenever she has an unusually hard problem at finding the right coins, she takes out her phone and calls you. Can you please help her not get beaten up by angry merchants?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a single line containing two integers n and c. n is the number of coin and note values and c is the amount of money that must be spent. The next line consists of n distinct integers $v_1, \ldots, v_n$ describing the coin/note values in increasing order. You may assume that a coin of value 1 is always available, and that Lea has as many of all the coins/notes as she needs.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is a space-separated sequence of n integers $a_1, \ldots, a_n$, where a_i means that the optimal solution uses exactly a_i coins/notes of value v_i.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 100$
- $1 \leq c \leq 10^5$
- $1 \leq v_i \leq 10^4$ for all $1 \leq i \leq n$

Sample Input 1	Sample Output 1
2 6 29 1 2 4 5 6 8 4 43 1 6 7 13	Case #1: 0 0 0 1 0 3 Case #2: 0 5 0 1

Sample Input 2

```
5
4 780
1 6 35 97

9 827
1 15 17 26 64 70 79 88 98

5 598
1 21 45 64 68

2 756
1 55

5 507
1 23 33 35 58
```

Sample Output 2

```
Case #1: 4 0 0 8
Case #2: 1 0 0 0 0 2 0 0 7
Case #3: 0 1 1 3 5
Case #4: 41 13
Case #5: 0 0 2 1 7
```

Problem B
Bracelets

Lea's neighbours have two daughters: Lara and Sarah. Last Christmas, Lea bought two nice silver bracelets for them. She had hoped they would like them, but they reacted as every child does: Each of them wanted to have the other's bracelet, but neither of them was willing to trade. Lea soon realized that she should have bought absolutely identical gifts.

Luckily, she had an idea how to fix this. The bracelets are built from smaller segments, each of them having a tiny picture on it. Lea is able to remove any of the segments, but by doing so she destroys them. Therefore, she may not exchange segments, but only remove some of them. She quickly figured out which segments to remove that both bracelets look the same. But now, weeks later, she still worries whether she could have removed other segments in a way that the remaining bracelets would have been longer.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line. Each test case consists of two lines containing one string a and b, respectively, describing the bracelets. The strings contain only the lower-case characters "a" to "z", representing the different pictures on the segments. Note that you may rotate the bracelets or turn them around, for instance "abcd" describes the same bracelet as "bcda" and "dcba" do. Also, the given bracelets may have different lengths.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is the maximum length of the identical bracelets after removing some segments. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq |a|, |b| \leq 250$

Sample Input 1	Sample Output 1
2 abcdef dcbafe abcdd dbac	Case #1: 6 Case #2: 3

Sample Input 2	Sample Output 2
17 qbt aoygjd zjj ohl plk kceplsidfw gskztvi cowv cyico dvu opzohlhojv lrv ossvcg khfvkbelkt qkoybfurhb aodnquw hdiin omnerhn tlhah extpcafhg fnn jkidv lypb ciymmmwc oturhkf vwqwl zhmrvh iguft glf xvigra gpxz hmgblxdgw qlmp qxxxxsvtes	Case #1: 0 Case #2: 0 Case #3: 3 Case #4: 1 Case #5: 0 Case #6: 2 Case #7: 1 Case #8: 3 Case #9: 2 Case #10: 3 Case #11: 0 Case #12: 1 Case #13: 0 Case #14: 0 Case #15: 1 Case #16: 2 Case #17: 1

Problem C
Packing Cases

Just recently, during Lea's visit at her uncle's house, she was reminded that while some people are quite tall, sadly she is not. She could not even reach the glasses that were stored in the topmost shelf in the kitchen. Luckily for her, there were a lot of packing cases lying around and she could use them to build a tower and then climb on it to reach the glasses.

Building such a tower is of course a very shaky endeavour, and Lea does not want to fall. So she imposed the following restriction on the tower: Given two packing cases a and b with dimensions x_a, y_a, z_a and x_b, y_b, z_b, case a may only be stacked onto case b if $x_a < x_b$ and $y_a < y_b$. Please remember that a case can be rotated to fit that restriction.

Lea now has to figure out whether it is possible to reach the desired height if she stacks the cases optimally, or not.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with two integers, h and n, the height the tower should reach and the number of case types, n lines follow. The i-th line describes the i-th case layout and contains three integers x_i, y_i, z_i. Lea has exactly 5 Boxes of each type at her disposal.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is either "yes" if Lea can build a tower of height at least h according to the constraints, or "no" if it is not possible to do so. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 1000$
- $1 \leq x_i, y_i, z_i \leq 40000$
- $1 \leq h \leq 5 \cdot 10^6$

Sample Input 1	Sample Output 1
11 9 1 5 4 3 7 2 4 2 2 3 1 5 8 2 3 5 5 2 2 3 6 3 4 1 1 5 2 4 3 1 3 8 2 5 2 2 4 5 1 10 4 2 1 5 2 5 3 4 1 1 5 2 4 6 3 1 2 1 1 5 3 5 4 2 7 1 3 1 3 7 2 1 5 4 6 5 6 2 4 5 4 1 1 3 4 3 6 3 2 6 2 8 2 1 6 6 2 2 2	Case #1: no Case #2: yes Case #3: yes Case #4: yes Case #5: no Case #6: yes Case #7: yes Case #8: no Case #9: yes Case #10: yes Case #11: no

Problem D
Poker

As you know by now, Lea has always been interested both in games and statistics. Lately, she came to realize that there is no game as good at combining these two as Poker. She began reading up on strategy - soon things like “short stack”, “big stack”, “sharks” etc. began making sense to her. Her iron will to win awakened as she delved headfirst into the world of competitive poker.

She decided she would make a grand entrance to the scene by winning as much money as she could in a single month. This would be easy - she expected to be able to win all amateur tournaments easily with her newfound knowledge. Today, she had a look at the timetable for all the different amateur poker tournaments that would take place over the next months. Unfortunately, some of them overlapped, so she would not be able to win all of them. Can you tell her how much money she can win without going to overlapping tournaments?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing n, the amount of tournaments that take place in the next few months. n lines follow, one for each tournament. Each line consists of 3 numbers, a b p, where the tournament starts on day a, ends on day b and has a prize of p € for the winner. Two tournaments overlap even if the first one ends on the same day the second one starts.

Output

For each test case, print a line containing “Case #i: x” where i is its number, starting at 1 and x is the most amount of money Lea can win from all the tournaments (given that she wins each tournament she attends). Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 5000$
- $1 \leq a \leq b \leq 5000$
- $1 \leq p \leq 10000$

Sample Input 1	Sample Output 1
2 3 1 5 1000 10 15 1000 2 12 3000 4 1 1 500 2 5 500 3 7 1000 6 7 600	Case #1: 3000 Case #2: 1600

Sample Input 2	Sample Output 2
5 3 6 9 10 9 13 8 10 15 8 4 3 8 6 7 11 10 5 9 10 8 11 9 3 9 12 5 8 13 8 1 6 6 3 8 11 10 1 5 6 5 10 11 4 2 7 7 4 7 8 3 8 5 1 6 6	Case #1: 18 Case #2: 10 Case #3: 14 Case #4: 16 Case #5: 8

Problem E
Escaping the Paradox

During a recent archaeological trip, Lea discovered a vast complex of underground tunnels connecting graves of an ancient civilisation. She spent a long time wandering around and drawing a map that captures all the graves and connections. Right now, she is deep below the surface, and has finally reached the last unexplored chamber, which is, disappointingly, completely empty. "Must have been budget cuts" she is thinking, when suddenly a distant feeling of danger makes her feel uneasy. Only an instant later, she knows with absolute certainty the source of the feeling: It is herself. Or, her future self, to be exact.

She does not yet know why she would want to use a time machine to come back here from the future, but basic knowledge of time travel is of course to never interact with yourself from another time. Luckily, she knows exactly where her future self is right now: Only one of the caves is big enough for the time machine. Now she has to figure out how she can escape to the surface without any chance of running into her clone. Her first decision is to always run towards the surface. Of course this problem is not that hard, but Lea also wants to take as many objects of the ancient civilization with her as possible. Luckily, her map includes a list of the objects in each room. How many objects can she take with her on her way to the surface while making sure there is no way to run into herself?

Lea and her future self need the same time to move the same distance and Lea has to take a way so that it is never possible for her and her future self to be in the same location at the same time. In particular Lea has to exit to the surface before her future self can do so. Furthermore, tunnels are undirected for future Lea, but Lea can only move in direction of the surface, that is, to graves with smaller indices.

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with three integers n, m, and g, the number of graves n, the number of tunnels m and the grave g where Lea's future self starts. Graves are indexed from 1 to n where n is the grave where Lea is right now. The following line contains n integers $o_1 \dots o_n$, o_i is the number of objects in grave i. m lines follow describing the tunnels, the j-th line contains three integers x_j y_j l_j. x_j and y_j are either grave indices or 0 (the surface) meaning that there is a tunnel between x_j and y_j of length l_j.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is either the maximal number of objects Lea can take with her on the way to the surface (may be 0) or "impossible" if she cannot escape her future self. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 500$
- $1 \leq m \leq 5000$
- $1 \leq g \leq n$
- $0 \leq o_i \leq 50$ for all $1 \leq i \leq n$
- $1 \leq l_j \leq 50$ for all $1 \leq j \leq m$
- $0 \leq x_j, y_j \leq n$ for all $1 \leq j \leq m$
- The graph is connected.

Sample Input 1

```
2
3 3 2
1 1 1
0 1 1
1 3 1
1 2 2

3 3 2
1 1 1
0 1 1
1 2 1
1 3 2
```

Sample Output 1

```
Case #1: 2
Case #2: impossible
```

Sample Input 2

```
4
2 6 1
2 0
2 1 6
1 0 6
1 2 6
2 1 6
2 2 7
0 0 4

4 5 2
2 2 3 3
0 4 4
4 1 2
1 3 6
3 2 6
3 4 7

4 7 3
2 2 1 0
0 1 6
1 2 7
0 4 6
4 3 6
0 4 7
4 4 2
0 3 6

4 7 3
3 4 0 3
2 4 4
2 3 6
4 1 3
1 0 6
1 0 7
1 1 6
0 0 2
```

Sample Output 2

```
Case #1: impossible
Case #2: 3
Case #3: impossible
Case #4: 6
```