· U C ·

UNIVERSIDADE DE COIMBRA

Departamento de Engenharia Informática

TIUP 2011

TORNEIO INTER-UNIVERSITÁRIO DE PROGRAMAÇÃO

Problema A	-	Matches and gaps
Problema B	-	Publish or perish
Problema C	-	Twenty-four
Problema D	-	We are tired of shortest paths!
Problema E	-	The problem of friendship

`http://mooshak.dei.uc.pt/~tiup2011`

Coimbra, 15 de Junho de 2011

Scientific Committee

- Carlos Fonseca
- Diogo Ferreira
- Filipe Araújo
- Francisco Pereira
- João Oliveirinha
- Luís Paquete

Lang.	**Compiler**	**Version**	Command Line	Ext.
C	gcc	4.1.2	gcc -ansi -Wall $file	.c
C	gcc	4.1.2	g++ -ansi -Wall $file	.cpp
Java	jdk	1.6.0_023	javac $file	.java

Problem A - Matches and gaps

Problem

Given an alphabet {"_", "A", "B", ..., "Z"}, where "_" denotes a space, consider two strings S1 and S2. We say that S1 and S2 have a *match* at position i if the i-th character, different from a space, is the same in both strings. In the example below, S1 and S2 have 4 matches at positions 1, 2, 3 and 6 ("P","E","N" and "O", respectively).

```
S1 = "PENACOVA"
S2 = "PENEDO"
```

We also say that a string S has a gap if it contains one or more consecutive spaces. In the following example, string S has two gaps (one with two spaces and another with one space).

```
S = "PE__NA_COVA"
```

An *alignment* is a pair of strings of the same length such that there exists no position where both strings have a space. For instance

```
"PENACOVA"
"PENEDO__"
```

is an alignment of strings S1 and S2 as given above, but the following is not because there is a space at the 5th position in both strings:

```
"PENA_COVA"
"PENE_DO__"
```

The following one is also not an alignment because both strings have different length:

```
"PENACOVA"
"_PENEDO"
```

Your goal is the following: Given two strings S1 and S2 with no spaces, transform them into an *alignment* such that the number of matches minus the number of gaps (that we call *score value*) is the maximum possible. You are allowed to add spaces to the left, to the right and within the strings.

For instance, given the following strings

```
S1 = "BANANA"
S2 = "ANANAS"
```

the optimal alignment is

```
"BANANA_"
"_ANANAS"
```

with 5 matches, 2 gaps, which gives a score value of 3.

Input

Each test case starts with the number of pairs of strings (n) in the first line. Then, the following $2n$ lines give the n pairs of strings.

Note that the input strings contain no spaces.

Output

For each pair of strings, print the maximum score value.

Constraints

Each string has at most 100 characters. Each test case can have up to 750 pairs of strings.

Sample Input

```
3
BANANA
ANANAS
PENACOVA
PENEDO
IMPOPULAR
POPULARUCHO
```

Sample Output

```
3
3
5
```

Problem B - Publish or perish

Problem

State-funded research units are increasingly under pressure to publish in scientific journals, in order to secure more funding in the future. Therefore, the director of a Research Unit has asked members to propose project ideas for the current year, indicating how much funding they would require to conduct the research and how many papers they would publish as a result. Each member is allowed to submit several proposals but, in order to keep as many members active as possible, at most one project per member will be funded. Members that do not receive funding are not expected to publish any paper.

Now, the director must decide which member projects to fund. The Research Unit's budget is limited, and the goal is to maximise the number of papers published as a result. Any funding which is not allocated to projects will have to be returned to the funding agency.

Input

There is a single test case in an input file. The first line contains two integers n and m corresponding to the number of researchers in the Unit and the Unit's budget in k€. The next line contains n positive integers $p_1, p_2, \ldots, p_n$ representing the number of proposals submitted by each member. Finally, for each proposal of each member, the amount of funding required and the number of papers promised follow on a separate line. The first p_1 lines correspond to the projects of the first member, the next p_2 lines correspond to the projects of the second member, and so on.

Output

The output for each test case is a line containing a single integer, indicating the maximum number of papers that can be obtained by funding at most one project per member.

Constraints

The maximum budget considered is 5000 k€. Research units have $n \leq 200$ researchers. Each researcher submits at least 1 proposal and at most 10.

Sample Input

```
5 20
3 4 2 1 3
10 3
5 10
8 10
3 3
1 18
5 10
10 3
10 8
8 7
1 5
4 2
11 11
5 5
```

Sample Output

```
45
```

Problem C - Twenty-four

Problem

Kids in primary school play a very simple, yet funny game: given four 1-digit numbers, get a result of 24 using the four basic operations, '−', '+', '×', '/'. For example, with the numbers 1, 2, 3, 4, you could do $1 \times 2 \times 3 \times 4 = 24$. With the numbers 2, 5, 6, 8, you could do $(8 - 5) \times (6 + 2) = 24$. However, with the numbers 7, 7, 8, 9 you have no solution to this problem.

Sometimes my kid asks me to play with him, and provide him with numbers. Unfortunately, we often struggle to find a solution, even though no solution exists. Can you help me on this?

You only need to read 4 1-digit numbers from the same input line and output "yes", if there is some solution, or "no", if no solution exists. You may find more than one input line. You should output "yes" or "no" for each line with 4 numbers, and should silently stop when the input ends.

Input

Each line will have 4 1-digit numbers between 0 and 9 separated by white space.

Output

For each line, output "yes" or "no".

Constraints

The input file has at most 1000 lines.

Sample Input

```
1 2 3 4
2 5 6 8
7 7 8 9
4 4 4 4
6 5 4 3
1 2 4 8
```

Sample Output

```
yes
yes
no
yes
yes
yes
```

Problem D - We are tired of shortest paths!

Problem

Let's face it, you love solving problems. There is some weird but quite nice appeal to being stuck in a room for hours torturing your brain just to achieve a solution.

Like you, Sarah loves programming competitions but she has one strict rule:

> "What happens during the programming contest stays at the programming contest."

After the contests, she goes back home but she never takes the shortest path. Why would she? Calculating shortest paths is something you do during contests. Instead, she would like to take the second shortest path.

Since you are in a programming contest right now, can you help Sarah find the path she is looking for? She wants you to know that a path that is the same length as the first shortest path is still a first shortest path and she does not care about those!

Input

The input consists of multiple test cases. Each one starts with a couple of numbers N and E. N is the number of places Sarah can visit, E is the number of one-way connections between such places.

The following E lines contain three numbers each, p, q and d which represent a connection between p and q with distance d. Please note that several connections between pairs of nodes are possible.

Each test case ends with two numbers S, F which represent Sarah's position and the finishing position.

Output

The output for each test case consists of a line containing the total distance traveled in the second shortest path. If there is no such path or if there are multiple first shortest paths you should print `not found`.

Consecutive test cases should be separated by a newline.

Constraints

$0 < N \leq 1000$
$N - 1 \leq E \leq 1500$
$0 \leq d \leq 100$
$0 \leq p, q, S, F \leq N - 1$

Sample Input

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

5 5
0 1 2
1 2 1
```

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

Sample Output

```
not found

8
```

Problem E - The problem of friendship

Problem

The summer is knocking on our door, and the preparations for the festivals have already started. A number of friends participated in a challenge, and some of them won beer tickets for a festival. However, not everything is party, and some thinking is needed in order to avoid problems between friends.

Since they are all friends, those who have spare tickets decide to share them with the others. However, they do not want to be unfair with one another, which raises some constraints. Firstly, nobody should receive more tickets than if they were distributed equally among those that did not get any. Secondly, there are different levels of friendship. If someone has X spare tickets, the maximum number of tickets that he or she can give to a friend is X/L, where L is their (integer) friendship level.

Taking this into account, can you help them so that they share as many spare tickets as possible with each other?

Input

The first line of input specifies the number of friends that have spare tickets (N) and the number of friends that want tickets (M). N lines follow, each describing the number of tickets that each friend has. The following lines contain three positive integers each: a number that represents the friend with tickets, a number that represents a friend that wants tickets, and the level of friendship between those two friends. The input ends when the end of file is reached.

Output

The output of your solution should be the maximum number of tickets that can be shared under the friendship constraints above.

Constraints

$2 < N + M <= 400$

Sample Input

```
3 5
88
58
61
1 4 2
1 7 1
1 8 3
2 4 2
2 5 2
2 7 3
2 8 1
3 8 2
```

Sample Output

```
152
```