South African Computer Olympiad
Final Round 2011
Day 2

Overview

Author(s)			
Problem	**highway**	**garbage**	**auction**
Source	highway.java highway.py highway.c highway.cpp highway.pas	N/A	auction.java auction.py auction.c auction.cpp auction.pas
Input file	stdin	garbage.in	stdin
Output file	stdout	garbage.out	stdout
Time limit	2 seconds	N/A	1 second
Memory limit	64MiB	N/A	64MiB
Number of tests	10	10	10
Points per test	10	10	10
Detailed feedback	No	No	Yes
Total points	**100**	**100**	**100**

The maximum total score is 300 points.

Highway Builder

Introduction

Bentopia has just gone through a long war with the Republic of Tiger. The Republic of Tiger systematically dropped Hunterbombs on all of Bentopia's transport networks, leaving the N cities in Bentopia completely cut off from one another.

Ben, the fearless dictator of Bentopia, has a plan to rebuild the transport networks. He wants to build highways between the cities such that one can drive from any city to any other city (possibly via intermediate cities) using the highways.

The cities are numbered from 1 to N. Luckily, there are expert highway builders in city 1; unfortunately, they are not that smart at planning which highways to build, and in what order to build them in. To build a highway from city a to city b, there must already be a route to city a from the builders' current location using highways already built. When the builders build the highway from city a to city b they end up at city b; afterwards, they can travel along the highway in either direction.

Task

Before building a highway, the builders call Ben and ask if they can build a highway from city a to city b. Ben must then work out if they can travel to city a. Ben also wants to prevent building unnecessary highways, so he checks whether there is already a route between city a and city b. If the builders can travel to city a and there is no existing route between a and b, he says "yes". The workers then go and build the highway. If Ben says "no", then they suggest another highway.

Your task is to help Ben decide whether to say "yes" or "no".

Example

In the example Bentopia has 6 cities. Highways are built such that you can reach every city but city 6. The builders' suggestions and Ben's responses are described in the table below, and depicted in Figure 1.

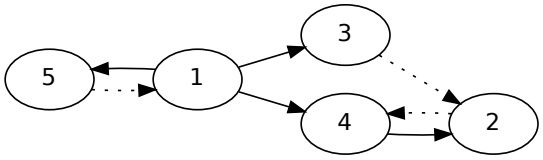

Figure 1: The circled numbers represent cities. A solid edge represents a highway that will be built. A dashed edge represents a highway that will not be built. The arrows represent the direction in which the highways are built (or proposed).

From	To	Decision	Reason
5	1	NO	Can't reach city 5
1	3	YES	
2	4	NO	Can't reach city 2
1	4	YES	
4	2	YES	
3	2	NO	Can already reach city 2 from city 3
1	5	YES	

Input (stdin)

The first line of input contains two space-separated integers, N and M. The next M lines each contain two integers, a_i and b_i.

Sample input

```
5 7
5 1
1 3
2 4
1 4
4 2
3 2
1 5
```

Output (stdout)

The output contains M lines, each containing Ben's answer to the corresponding suggestion: either `YES` or `NO`.

Sample output

```
NO
YES
NO
YES
YES
NO
YES
```

Constraints

- $2 \leq N \leq 10\,000$
- $N - 1 \leq M \leq 40\,000$
- $1 \leq a_i, b_i \leq N$
- $a_i \neq b_i$

Additionally, in 50% of the test cases:

- $2 \leq N \leq 10$
- $M \leq 100$

Time limit

2 seconds.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.

Garbage Detection

Introduction

Somebody has been messing with Bruce's computer: they have scrambled some of his documents, filling them with nonsense.

Bruce is too busy to sort out the documents himself, so he has asked you to help him. To make this more difficult, some of the nonsense looks superficially like English text.

Task

Given a number of pieces of text, some of which are fragments of genuine English-language documents, and the rest of which are nonsense, you must determine which are which.

In a single input, each nonsensical fragment is randomly generated in the same way. The genuine text—both in the files and in the input to the nonsense generators—is extracted from published works.

Example

Consider the following fragments:

1. "Computer Olympiad."
2. "Rtocnodoaces, gduc boq?"

The first is English, while the second is nonsense.

Input (garbage.in)

Each input file contains multiple text fragments. The first line of the input file contains N, the number of fragments. The N lines each contain an integer W, followed by W space-separated words of text.

For ease of programming, all non-ASCII characters have either been converted to ASCII or removed, and all strings of whitespace have been replaced with a single space character.

Sample input

```
2
2 Computer Olympiad.
3 Rtocnodoaces, gduc boq?
```

Output (garbage.out)

The output file consists of N lines. Each line should contain a single string, either `DOCUMENT` to indicate that the corresponding text fragment in the input file contains text from a real document, or `GARBAGE` to indicate that it does not.

Sample output

```
DOCUMENT
GARBAGE
```

Format checking

When you upload an output file, it will be checked to ensure that it has the correct number of lines, and that each line is either `DOCUMENT` or `GARBAGE`.

Scoring

If a file contains the correct answer for C fragments out of N, it will score

$$\max\left\{0, \left(\frac{C}{N} - \frac{1}{2}\right) \times 20\right\},$$

rounded down. Note that this implies that more than half of the answers in a file must be correct to get any points for that file.

Hints

- You can find a list of English words in the `/usr/share/dict/words` file. This may help you solve some of the input files (but not others).
- The nonsense fragments in different files have not all been generated in the same way. You should examine each file.
- Nevertheless, all the nonsense generators use some form of randomness.

Auction

Introduction

Fred the Manic Storekeeper's sales have not been that great recently thanks to the economic downturn. To help supplement his income Fred has started auctioning off toys on the weekends.

Fred has N different types of toys (numbered from 1 to N) from his store which he is putting up for auction. In fact he has K of each type of toy in stock, and is putting all K of each type up for auction.

Fred has noticed that bidders tend to only bid on a few of the types of toys up for auction, so he has thought up some rules to force bidders to bid on a wider range of toys. Bidders must place bids on pairs of toy types, and they place bids in turn. The first bidder can bid on any pair of toy types, a_1 and b_1, for c_1 rand. After that, bidder number i bids c_i rand on toy types a_i b_i, where a_i must have previously been bid on and b_i must have had no bid on it yet. Fred stops the auction once every type of toy has been bid on. The rules imply that there are exactly $N - 1$ bids.

Fred is allowed to choose which bids to accept and he wants to make as much money as possible.

Task

Your task is to help maximise the amount of money Fred makes. If Fred accepts the bid of bidder number i, then he makes c_i rands and gives the bidder one of toy a_i and one of toy b_i. Remember that Fred only has K of each type of toy in stock.

Example

In the example input there are 7 different types of toys up for auction, with 2 of each toy in stock. The maximum amount of money that Fred can make is R 13. He does this by accepting the $3^{\text{rd}}, 4^{\text{th}}, 5^{\text{th}}$ and 6^{th} bids. These are bids are for R 4, R 3, R 2 and R 4 respectively. This is the only way for Fred to make R 13.

Input (stdin)

The first line of input contains two space-separated integers, N and K. The next $N - 1$ lines each contain three integers, a_i, b_i and c_i. For every line but the first it is guaranteed that toy type a_i has appeared previously and that toy type b_i has not.

Sample input

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

Output (stdout)

The output contains a single line containing a single integer, the maximum amount of money Fred can make.

Sample output

```
13
```

Constraints

- $1 \leq K \leq N \leq 20\,000$
- $1 \leq a_i, b_i \leq N$
- $1 \leq c_i \leq 1\,000$
- $a_i \neq b_i$

Additionally:

- in at least 10% of the test cases $K = N$
- in at least 20% of the test cases $N \leq 5$
- in at least 40% of the test cases $K = 2$

Time limit

1 second.

Detailed feedback

Detailed feedback is enabled for this problem.

Scoring

A correct solution will score 100% while an incorrect solution will score 0%.