All the programs were tested on IBM PC Pentium, 100 MHz computers.

THE RESULTS OF A BASKETBALL TOURNAMENT (Latvia)
35 points, run time limit – 60 sec.

N teams played one game with each other in a basketball tournament. There were no tie results.

At the end of the tournament there was a necessity to rank all the teams from the strongest to the weakest, using the following criteria:

Criterion A: *The team which has more wins in the tournament takes a higher place.*

If more than one team has the same number of wins, the results of the "micro-tournament" of these teams are considered – only the results of the games played between these teams are taken into account:

Criterion B: *The team which has more wins in a micro-tournament takes a higher place.*

If there are teams with the same number of wins in the micro-tournament, then

Criterion C: *The team which has a greater difference between the collected and lost points in the micro-tournament takes a higher place.*

In case this number is the same for several teams, then

Criterion D: *The team which has collected more points in the micro-tournament takes a higher place.*

You are to write a program which ranks all teams in the right order for the given game results, beginning with the first and finishing with the last place. It is known that the rank is unique for the given data.

Let us assume that 6 teams ABAVA, DAUGAVA, GAUJA, IMULA, LIELUPE and VENTA have participated in the tournament, and the results of the games are given in the table (wins are specified in bold):

name							wins	place
ABAVA		75:143	131:137	**92:51**	131:135	40:109	**1**	6
DAUGAVA	**143:75**		131:137	82:136	**108:80**	**112:58**	**3**	
GAUJA	**137:131**	**137:131**		**64:24**	45:128	56:148	**3**	
IMULA	51:92	**136:82**	24:64		76:106	**104:87**	**2**	5
LIELUPE	**135:131**	80:108	**128:45**	**106:76**		108:146	**3**	
VENTA	**109:40**	58:112	**148:56**	87:104	**146:108**		**3**	

As we can see, using the number of wins in the tournament (criterion A), it is possible to determine the rank of just two last teams – IMULA and ABAVA. To find the right order of the other four teams, it is necessary to consider the micro-tournament of these teams:

name					wins	points collected	points lost	difference
DAUGAVA		131:137	**108:80**	**112:58**	**2**	351	275	+76
GAUJA	**137:131**		45:128	56:148	**1**	238	407	–169
LIELUPE	80:108	**128:45**		108:146	**1**	316	299	+17
VENTA	58:112	**148:56**	**146:108**		**2**	352	276	+76

Having taken into account the number of wins in this micro-tournament (criterion B), it is possible just to say that the teams DAUGAVA and VENTA will take higher places than GAUJA and LIELUPE.

To determine which of the teams DAUGAVA or VENTA takes a higher place, it is not enough to take into account the difference between the collected and the lost points (criterion C), because this number is the same for both teams (+76). Then remains the last index - points, collected in the micro-tournament (criterion D). The team VENTA has collected more (352) than DAUGAVA (351), so VENTA takes a higher place than DAUGAVA.

To determine which of the teams GAUJA or LIELUPE takes a higher place, it is enough to consider the difference of the collected and the lost points (criterion C), because this value is greater for the team LIELUPE (+17).

So, the final rank of teams is: VENTA, DAUGAVA, LIELUPE, GAUJA, IMULA, ABAVA.

Input Data

There is one whole positive number in the first line of the input file BASKET.DAT – the number of teams in the tournament n ($2 \le n \le 300$). The results of the games are presented in the following lines. There is one line in the file for each game. In each line the result of one game is given in the following form:

<x_name>–<y_name> <x_collected_points>:<y_collected_points>

There are no additional blanks in the rows (there is a space character only between the team *y* name and points collected by the team *x*).

Each team name contains only the capital letters of the Latin alphabet and the length of the name does not exceed 20 characters. The number of the collected points by a team in any game does not exceed 150.

Output Data

The result file BASKET.REZ must contain *n* rows. The name of one team must be written in each row of the file, starting with the first position. In the *i*-th the row of the file must be the name of the team which took the *i*-th place in the tournament.

Example

INPUT
(file BASKET.DAT)

```
6
GAUJA–LIELUPE 45:128
IMULA–DAUGAVA 136:82
IMULA–VENTA 104:87
ABAVA–IMULA 92:51
DAUGAVA–LIELUPE 108:80
ABAVA–GAUJA 131:137
DAUGAVA–VENTA 112:58
DAUGAVA–ABAVA 143:75
IMULA–GAUJA 24:64
VENTA–ABAVA 109:40
GAUJA–VENTA 56:148
ABAVA–LIELUPE 131:135
VENTA–LIELUPE 146:108
IMULA–LIELUPE 76:106
DAUGAVA–GAUJA 131:137
```

OUTPUT
(file BASKET.REZ)

```
VENTA
DAUGAVA
LIELUPE
GAUJA
IMULA
ABAVA
```

All the programs were tested on IBM PC Pentium, 100 MHz computers.

A SUBSEQUENCE (Poland)
20 points, run time limit – 10 sec.

We are looking for the longest sub-sequence of the given sequence of integers whose sum of elements is divisible by 3.

Write a program which:

- reads the sequence of integers from the text file SEQ.IN;
- computes the length of the longest sub-sequence whose sum of elements is divisible by 3;
- writes the output in the text file SEQ.OUT;

Input Data

There is one integer number n ($1 \le n \le 10000$) in the first line of the text file SEQ.IN. In each of the following n lines there is an element of the sequence $0 \le a_i \le 20000$ ($i = 1..n$).

The sequence is written in the input file one element per line.

Output Data

Write the length of the longest sub-sequence in the first line of the output file SEQ.OUT.

Example

INPUT (file SEQ.IN)	OUTPUT (file SEQ.OUT)
7 10 6 7 12 4 7 22	5

All the programs were tested on IBM PC Pentium, 100 MHz computers.

A MAZE WITH DOORS (Lithuania)
45 points, run time limit – 60 sec.

Let us imagine a man lost in a rectangular maze having m x n cells. Each cell is one of the following:

- a wall. It is not possible to walk into this kind of a cell;
- the floor. One can freely walk through this cell;
- a door. It is just like a wall, until a corresponding switch (see next section) is not turned on. When the switch is turned on, you can walk through the door just like trough the floor. However, the door stays open for only 20 moves after the switch has been turned on. Then the switch has to be turned on again to re-open the door. If the door is about to close while you are inside its cell, you cannot enter this cell. Note, that if you enter the switch cell when the corresponding door is still open, the closing counter of that door is reset to 20 moves again. There are no more than 20 doors in the maze.
- the floor with a switch. A switch is connected with some particular door, and when a man enters this cell, the corresponding door remains opened for next 20 moves. Each door has exactly one switch and vice versa (i.e. there are neither two switches for one door nor door without a switch or vice versa).

Write a program to find the shortest way out of the maze.

Input Data

The first line of the input file MAZE.DAT contains the co-ordinates (x, y, where $1 \le x \le m$, $1 \le y \le n$) of a man lost in the maze (this cell of the maze is always the floor without a switch).

The second line contains the numbers m and n ($m, n \le 50$) – the width and the length of the maze.

There are m numbers in each of the following n lines. Each number describes one cell in the maze:

–1	*– Wall*
0	*– Floor*
1-20	*– Door*
101-120	*– the Floor with a switch* (the number j means that this switch is connected with the door number j – 100)

Output Data

One integer number – the length of the shortest way out of the maze (or the string "NOWAY" if such a way does not exist) must be written to the output file MAZE.REZ.

Example

INPUT
(file MAZE.DAT)

```
2 2
8 6
-1 -1 -1 -1  -1 -1 -1 -1
-1  0  0  0   7  0  0  0
-1  0 -1 -1  -1 -1 -1 -1
-1  0 -1 -1 107 -1 -1 -1
-1  0  0  0   0 -1 -1 -1
-1 -1 -1 -1  -1 -1 -1 -1
```

OUTPUT
(file MAZE.REZ)

```
21
```