Problem (A)

5676. STONE GAME

Problem code: RESN04

Alice and Bob play the following game: There are N piles of stones with Si stones in the ith pile. Piles are numbered from 1 to N. Alice and Bob play alternately, with Alice starting. In a turn, the player chooses any pile i which has at least i stones in it, and removes exactly i stones from it. The game ends when there is no such pile. The player who plays last wins the game. Assuming Alice and Bob play optimally, who will win the game?

### Input

The first line contains the number of test cases T (<= 100). There follow 2T lines, 2 per test case. The first line of each test case contains N (<= 100), the number of piles. The second line contains N space separated integers, specifying the number of stones in pile 1, pile 2, ..., pile N. There will be at least 1 and at most 1000 stones in any pile.

### Output

Output T lines, one per test case. For each test case, output "ALICE" if Alice wins the game, and "BOB" if Bob wins the game.

### Example

**Input:**  
2  
1  
1  
2  
1 1  
  
**Output:**  
ALICE  
ALICE

Problem (B)

3386. Contest System Quality (QUALITY)

Write a program to score a small, three-problem programming contest. Each input line contains six space-separated integers representing raw score data. The first three integers are in the range 0 . . . 100000. They represent seconds taken to solve the first, second, and third problems, respectively. Zero seconds indicates that a problem has not been solved. The last three integers are in the range 0 . . . 100, representing the attempts taken to solve the first, second, and third problems, respectively. Every failed attempt is penalized with 20 minutes, but only for problems that are eventually solved.  
Each output line should begin with the string team, followed by a single space, the input line number, a colon, a single space, the number of solved problems, a comma, a single space, and the total number of seconds including penalties it took for the solved problems.

**Input:**

0 777 0 4 1 1

1 1 1 1 1 1

**Output:**

team 1: 1, 777

team 2: 3, 3

# Problem (C)

# 9754. Egypt (SCPC11D)

A long time ago, the Egyptians figured out that a triangle with sides of length 3, 4, and 5 had a right angle as its largest angle. You must determine if other triangles have a similar property.

**Input**

Input represents several test cases, followed by a line containing 0 0 0. Each test case has three positive integers, less than 30,000, denoting the lengths of the sides of a triangle.

### Output

### For each test case, a line containing "right" if the triangle is a right triangle, and a line containing "wrong" if the triangle is not a right triangle.

### Example

**Input:**

6 8 10

25 52 60

5 12 13

0 0 0

**Output:**

right

wrong

right

Problem (D)

95. Street Parade

Problem code: STPAR

For sure, the love mobiles will roll again on this summer's street parade. Each year, the organizers decide on a fixed order for the decorated trucks. Experience taught them to keep free a side street to be able to bring the trucks into order.   
  
The side street is so narrow that no two cars can pass each other. Thus, the love mobile that enters the side street last must necessarily leave the side street first. Because the trucks and the ravers move up closely, a truck cannot drive back and re-enter the side street or the approach street.   
  
You are given the order in which the love mobiles arrive. Write a program that decides if the love mobiles can be brought into the order that the organizers want them to be.

### Input

There are several test cases. The first line of each test case contains a single number n, the number of love mobiles. The second line contains the numbers 1 to n in an arbitrary order. All the numbers are separated by single spaces. These numbers indicate the order in which the trucks arrive in the approach street. No more than 1000 love mobiles participate in the street parade. Input ends with number 0.

### Output

For each test case your program has to output a line containing a single word yes if the love mobiles can be re-ordered with the help of the side street, and a single word no in the opposite case.

### Example

**Sample input:**

5

5 1 2 4 3

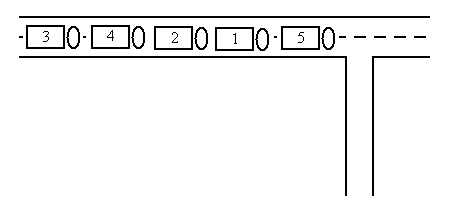
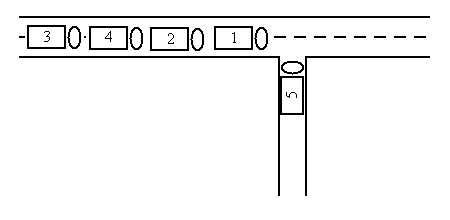
0

**Sample output:**

yes

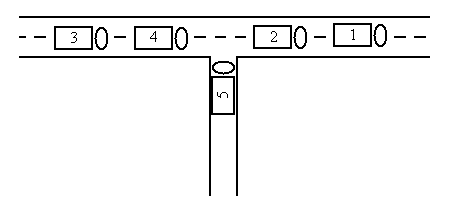
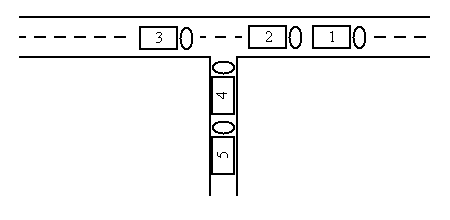
### Illustration

The sample input reflects the following situation:



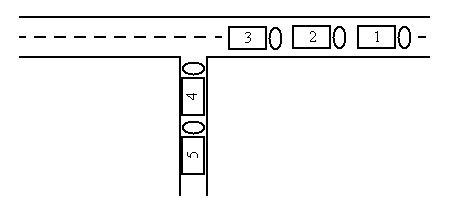
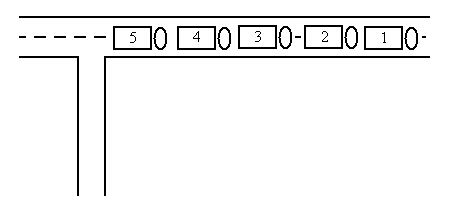
2

1



4

3



6

5

# Problem (E)

# 2523. Mispelling

## Problem code: GNY07A

Misspelling is an art form that students seem to excel at. Write a program that removes the nth character from an input string.

### Input

The first line of input contains a single integer N, (1 ≤ N ≤ 1000) which is the number of datasets that follow.

Each dataset consists of a single line of input containing M, a space, and a single word made up of uppercase letters only. M will be less than or equal to the length of the word. The length of the word is guaranteed to be less than or equal to 80.

### Output

For each dataset, you should generate one line of output with the following values: The dataset number as a decimal integer (start counting at one), a space, and the misspelled word. The misspelled word is the input word with the indicated character deleted.

### Example

**Input:**

4

4 MISSPELL

1 PROGRAMMING

7 CONTEST

3 BALLOON

**Output:**

1 MISPELL

2 ROGRAMMING

3 CONTES

4 BALOON

Problem (F)

8847. Google is Feeling Lucky

Problem code: FEELUCK

Google is one of the most famous Internet search engines. The question is, when one types some keywords and presses "I'm feeling lucky" button, which web page will appear? Google does a lot and comes up with excellent approaches to deal with it. In this simplified problem, let us just consider that Google assigns every web page an integer-valued relevance. The most related page will be chosen. If there is a tie, all the pages with the highest relevance are possible to be chosen.

Your task is simple, given 10 web pages and their relevance. Just pick out all the possible candidates which will be served to the user when "I'm feeling lucky".

### Input

### The input contains multiple test cases. The number of test cases is in the first line of the input file.For each test case, there are 10 lines, describing the web page and the relevance. Each line contains a character string without any blank characters denoting the URL of this web page and an integer ****Vi**** denoting the relevance of this web page. The length of the URL is between 1 and 100 inclusively. (1 <= ****Vi**** <= 100)

### Output

For each test case, output several lines which are the URLs of the web pages which are possible to be chosen. The order of the URLs is the same as the input. Please look at the sample output for further information of output format.

### Example

**Input:**

2

**Output:**

Case #1:

www.alibaba.com

www.bad.com

acm.university.edu.cn

Case #2:

www.alibaba.com

www.youtube.com 1

www.google.com 2

www.google.com.hk 3

www.alibaba.com 10

www.taobao.com 5

www.bad.com 10

www.good.com 7

www.fudan.edu.cn 8

www.university.edu.cn 9

acm.university.edu.cn 10

www.youtube.com 1

www.google.com 2

www.google.com.hk 3

www.alibaba.com 11

www.taobao.com 5

www.bad.com 10

www.good.com 7

www.fudan.edu.cn 8

acm.university.edu.cn 9

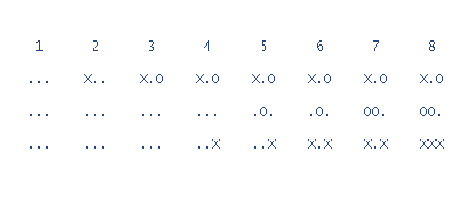
acm.university.edu.cn 10

# Problem (G)

# 1161. Tic-Tac-Toe ( I ) (TOE1)

Tic Tac Toe is a child's game played on a 3 by 3 grid. One player, X, starts by placing an X at an unoccupied grid position. Then the other player, O, places an O at an unoccupied grid position. Play alternates between X and O until the grid is filled or one player's symbols occupy an entire line (vertical, horizontal, or diagonal) in the grid.

We will denote the initial empty Tic Tac Toe grid with nine dots. Whenever X or O plays we fill in an X or an O in the appropriate position. The example below illustrates each grid configuration from the beginning to the end of a game in which X wins.

Your job is to read a grid and to determine whether or not it could possibly be part of a valid Tic Tac Toe game. That is, is there a series of plays that can yield this grid somewhere between the start and end of the game?

### Input

The first line of input contains N, the number of test cases. 4N-1 lines follow, specifying N grid configurations separated by empty lines.

### Output

For each case print "yes" or "no" on a line by itself, indicating whether or not the configuration could be part of a Tic Tac Toe game.

### Example

**Input:**

2

**Output:**

yes

no

X.O

OO.

XXX

O.X

XX.

OOO

# Problem (H)

# 18149. Rama and Friends (GSHOP)

Mahesh and Ramashish are good friends. Each day Ramashish gives an array arr[0,1,2...,n-1] of size 'n' to Mahesh and asks him to modify it.  
Modifying an array means to execute the following operation exactly 'k' times:  
"Replace any array element 'x' by '-x' i.e multiply it by -1."  
Note that this operation can be performed on an array element zero, one or more times.  
But while modifying the array, Mahesh should also keep in mind that the sum of the elements of the final array should be **maximum possible**. Mahesh wants to go to sleep so he has to finish this job as soon as possible. Can you help him?

### Input

First line of the input contains the number of test cases 'T'.  
Then follow 2T lines describing the test cases(Two lines for every test case).  
For each test case, First line has two space separated integers 'n'(Number of elements in the array) and 'k'(Number of times to execute the operation described above) and the next line has 'n' space separated integers in non-decreasing order which are the array elements.  
**Constraints:**  
1<=T<=10.  
1<=n<=100.  
1<=k<=100.  
For each 0<=i<=n-1,-10^5<=arr[i]<=10^5.

### Output

Output T lines, one for each test case denoting the maximum possible sum which can be obtained after modifying the array.

### Example

**Input:**

1

3 1

-1 -1 1

**Output:**

1

# Problem (I)

# 6830. Soccer Bets (SBETS)

The teams have finished the group stage of the FIFA World Cup and the teams that are in the round of the last sixteen are known. My boss has all of the games analyzed and bets on the whole rest of the tournament by writing the outcome of each match on a single sheet of paper. It was my job to bring his bets to the next betting office and set 1000 dollar. Being nervous with so much cash in my pockets I fell over (I am a bit clumsy) and the bets got shuffled. So I don't know if a bet corresponds to the final match or the semi-final or something else.

I do not want to disappoint my boss, so I decided to place only one bet on the winner of the tournament. Everything I know is that in each round the teams that win (a team wins if it shoots more goals than the opposing team) are in the next round, the other teams are eliminated from the tournament. This is not true for the semi-finals where the losers also play for the third place. So we have in total 16 matches. Can you please tell me which team will win the World Cup based on the bets of my boss?

### Input

The first line of the input is the number of test cases c (1 ≤ c ≤ 100). Each test case consists of 16 lines describing the matches in random order. A match description looks as follows: t1 t2 g1 g2. t1 and t2 are the names of teams (abbreviated as exactly three uppercase letters), g1 and g2 (0 ≤ g1, g2≤ 10; g1≠ g2) are the goals of the two teams.

### Output

For each test case, print one line containing the team that will win the FIFA World Cup (based on the analysis of my boss which is always correct!).

**Input:**

1

ITA URU 2 0

ITA IRE 1 0

**Output:**

GER

ITA ARG 3 4

YUG ARG 2 3

GER CZE 1 0

ENG GER 3 4

ITA ENG 2 1

CAM COL 2 1

ENG CAM 3 2

ENG BEL 1 0

GER ARG 1 0

CZE CRC 4 1

NET GER 1 2

BRZ ARG 0 1

SPA YUG 1 2

ROM IRE 4 5

Problem (J)

1025. Fashion Shows (FASHION)

A fashion show rates participants according to their level of hotness. Two different fashion shows were organized, one for men and the other for women. A date for the third is yet to be decided ;) .

Now the results of both fashion shows are out. The participants of both the fashion shows have decided to date each other, but as usual they have difficulty in choosing their partners. The Maximum Match dating service (MMDS) comes to their rescue and matches them in such a way that that maximizes the hotness bonds for all couples.

If a man has been rated at hotness level x and a women at hotness level y, the value of their hotness bond is x\*y.

Both fashion shows contain **N** participants each. MMDS has done its job and your job is to find the sum of hotness bonds for all the couples that MMDS has proposed.

### Input

The first line of the input contains an integer **t**, the number of test cases. **t** test cases follow.

Each test case consists of 3 lines:

* The first line contains a single integer **N** (1 <= **N** <= 1000).
* The second line contains **N** integers separated by single spaces denoting the hotness levels of the men.
* The third line contains **N** integers separated by single spaces denoting the hotness levels of the women.

All hotness ratings are on a scale of 0 to 10.

### Output

For each test case output a single line containing a single integer denoting the sum of the hotness bonds for all pairs that MMDS has proposed.

### Example

**Input:**

2

2

1 1

3 2

3

2 3 2

1 3 2

**Output:**

5

15

Problem (K)

2178. He is offside! (OFFSIDE)

Hemisphere Network is the largest television network in Tumbolia, a small country located east of South America (or south of East America). The most popular sport in Tumbolia, unsurprisingly, is soccer; many games are broadcast every week in Tumbolia.

Hemisphere Network receives many requests to replay dubious plays; usually, these happen when a player is deemed to be offside by the referee. An attacking player is *offside* if he is nearer to his opponents’ goal line than the second last opponent. A player is not offside if

* he is level with the second last opponent or
* he is level with the last two opponents.

Through the use of computer graphics technology, Hemisphere Network can take an image of the field and determine the distances of the players to the defending team’s goal line, but they still need a program that, given these distances, decides whether a player is offside.

### Input

The input file contains several test cases. The first line of each test case contains two integers A and D separated by a single space indicating, respectively, the number of attacking and defending players involved in the play (2 <= A,D <= 11). The next line contains A integers Bi separated by single spaces, indicating the distances of the attacking players to the goal line (1 <= Bi <= 104). The next line contains D integers Cjseparated by single spaces, indicating the distances of the defending players to the goal line (1 <= Cj <= 104). The end of input is indicated by A = D = 0.

### Output

For each test case in the input print a line containing a single character: “Y” (uppercase) if there is an attacking player offside, and “N” (uppercase) otherwise.

### Example

**Input:**

2 3

500 700

700 500 500

2 2

200 400

200 1000

3 4

530 510 490

480 470 50 310

0 0

**Output:**

N

Y

N