SPOJ Problem Set (tutorial)

244. Internally Stable Sets

Problem code: INTSS

A weighted finite undirected graph is a triple $G = (V, E, w)$ consisting of vertex set V, edge set [IMAGE] , and vertex weighting function w such that [IMAGE] and [IMAGE] . For [IMAGE] and [IMAGE] , $N(u)$ and $N(K)$ will denote the neighboring vertex sets of u and K respectively, formally defined as:
[IMAGE]
A vertex set [IMAGE] satisfying [IMAGE] is called *internally stable* (also known as independent or anti-clique). In this problem you must find an internally stable set B such that $w(B) = max\{w(S)\}$, where S belongs to the set of all internally stable sets of that graph.

Input

t - the number of test cases [t <= 100]
n k - [n - number of vertices (2 <= n <= 200), k - number of edges (1 <= k <= $n*(n-1)/2$)]
then n numbers follows (wi - the weight of i-th vertex) [0 <= wi <= 2^31-1]
then k pairs of numbers follows denoting the edge between the vertices (si sj edge between i-th and j-th verices) [1 <= si, sj <= n]

Output

For each test case output *MaxWeight* - the weight of a maximum internally stable set of the given graph. [0 <= *MaxWeight* <= 2^31-1]

Example

```
Input:
2
5 6
10 20 30 40 50
1 2
1 5
2 3
3 4
3 5
4 5

4 4
10 4 10 14
1 2
2 3
3 4
4 1

Output:
70
20
```

Added by:	Roman Sol
Date:	2004-12-14
Time limit:	21s
Source limit:	50000B
Languages:	All
Resource:	ZCon

SPOJ Problem Set (tutorial)

360. Bottom Coder (Easy)

Problem code: BCEASY

Some of you may be familiar with the TopCoder (TM) contest. Its exact rules are not important for this problem, but know that the most important part of it is writing a program according to the given specification. Many times the contestant ends up with a program which would work perfectly - if only he could change a couple of characters (like, replacing "=" by "==" in C, etc.). Unfortunately, even the best programmers sometimes aren't able to spot these tiny but necessary changes until it's too late... and that's why we developed a brand-new BottomCoder training for them!
The idea is very simple - you're given a problem specification, a source code, and a list of permitted modifications. Your task is to find a modification which would cause the program to behave according to the specification.
Specification: "Write a program which outputs EXACTLY 42 asterisks and NOTHING more (e.g. NO end-of-line markers, like "\n", ...)"
The code you are supposed to modify:

```
int i, n=42;
main() {
  for(i=0; i<n; i--) {
    printf("*");
  }
}
```

As this is a really, really simple problem, you are only permitted to make exactly ONE of these modifications to the source: 1) Add one character to the source. 2) Delete one character from the source. 3) Replace one character in the source by a different one.
Moreover, it would be definitely too easy if we asked you to find just one solution, so you'll need to find TWO DIFFERENT solutions in order to obtain credit for this problem. (There are exactly three different solutions, so don't worry, it can be done!)

Input

There is no input for given problem.

Output

Your submission should consist of two parts. The first part should contain the first of your solutions. A single line with the letter "Q" follows. (Note that the letter Q is used as a separator. You will have to do without inserting the letter Q in at least one of your solutions :) After this line you should add your second solution.
You don't need to worry much about the exact formatting of your submission. The exact judging procedure will look as follows:
The first occurrence of the letter Q is found, the input is split into two parts. Any whitespace in each of the parts is removed. It is checked whether the two submissions differ and whether each of them was obtained from the original program by an allowed change. Each of your two submissions is compared to each of the three correct solutions.

Example

```
Output:
int i, n=42; main(
){ for(i=0; i<n; i--)   { printf("?"); } }
Q
int i, n=41; main() { for(i=0; i<n;i--) { printf("*"); } }
```

(syntactically valid (but incorrect) submission)

Added by:	Roman Sol
Date:	2005-05-13
Time limit:	1s
Source limit:	10000B
Languages:	TEXT
Resource:	IPSC 2005

SPOJ Problem Set (tutorial)

361. Bottom Coder (Hard)

Problem code: BCHARD

Some of you may be familiar with the TopCoder (TM) contest. Its exact rules are not important for this problem, but know that the most important part of it is writing a program according to the given specification. Many times the contestant ends up with a program which would work perfectly - if only he could change a couple of characters (like, replacing "=" by "==" in C, etc.). Unfortunately, even the best programmers sometimes aren't able to spot these tiny but necessary changes until it's too late... and that's why we developed a brand-new BottomCoder training for them!
The idea is very simple - you're given a problem specification, a source code, and a list of permitted modifications. Your task is to find a modification which would cause the program to behave according to the specification.
Specification: "Write a program which outputs a short English text mentioning our partner competition - IPSC. The text must consist of one or more English sentences and each sentence has to contain one or more English words (sequences of only upper-case characters) separated by spaces. Additionally, you may use certain punctuaction characters - namely ".!?,'". Try to obfuscate the program as much as possible." The code you are supposed to modify:

```
#include <stdio.h>

int rex[5];

void f3(int *a) {
  int i;
  for (i=0; i<5; i++) a[i]=0;
}

int f2(int *a) {
  int i;
  for (i=0; i<5; i++) if (a[i]!=0) return 0;
  return 1;
}

void f1(int *a) {
  int i;
  for (i=0; i<5; i++) {
    a[i]++;
    if (a[i]<100) break;
    a[i]-=100;
  }
  for (i=4; i>=0 && a[i]>=rex[i]; i--)
    if (a[i]>rex[i])
      f3(a);
}

void f4(int *a) {
  int i;
  for (i=0; i<5; i++) {
    a[i]--;
    if (a[i]>=0) break;
    a[i]+=100;
  }
```

```
  if (i>=5) for (i=0; i<5; i++) a[i]=rex[i];
}

void f7(int *a, int *b) {
  int c[5];
  f3(c); f3(a);
  while(!f2(b)) { f1(a); f4(b); f1(c); }
  while(!f2(c)) { f1(b); f4(c); }
}

void f9(int *a, int *b) {
  f1(a);
  while(!f2(b)) { f4(b); f1(a); }
}

void f8(int *a, int *b) {
  int c[5], d[5];
  f7(d, a);
  f3(a); f1(a);
  while(!f2(b)) { f7(c, d); f9(a, c); f4(a); f4(b); }
}

void f5(int *a, int *b) {
  int c[5], d[5];
  f7(d, a);
  f3(a); f1(a);
  while(!f2(b)) { f7(c, d); f8(a, c); f4(a); f4(b); }
}

void f10(int x) {
  int rpl[]=
{80, 125, 111, 18, 59, 88, 88, 28, 65, 98, 119, 103, 101, 79, 107, 2, 16,
92, 102, 123, 103, 84, 112, 78, 68, 98, 65, 37, 105, 85, 107, 13, 45, 9,
104, 81, 21, 31, 55, 110, 78, 66, 66, 3, 77, 63, 16, 105, 15, 123, 16, 84,
31, 96, 4, 82, 82, 122, 68, 115, 35, 73, 3, 108, 115, 83, 15, 19, 31, 99, 5,
123, 24, 65, 36, 15, 75, 84, 4, 2, -1};

  int i;
  int a[5], b[5], c[5];

  if (x<100000000 || x>200000000) return;
  x--;
  f3(rex); rex[4]=1;
  for (i=0; rpl[i]!=-1; i++)
  {
    f3(a); a[0]=i+1;
    f3(b); f1(b); f3(c); f1(b); f1(b);
    f1(c); f1(b); f5(a, b);

    f1(c);
    while(!f2(a))
    {
      f3(b); b[0]=x%100; b[1]=x/100;
      f4(a); f8(c, b);
    }
    rpl[i]^=c[1];
    printf("%c", rpl[i]);
  }
  printf("\n");
}
```

```
int main()
{
  f10(47);
}
```

As you can see, the coder made almost everything according to the specification :) You're only allowed to alter one number in the source code - namely the number 47 on line 98 (the argument of function "f10" called from "main"). You can replace it by any integer between 100‘000‘000 and 200‘000‘000 inclusive.

Input

There is no input for given problem.

Output

Your submission should consist of two lines. The first line should contain the value of the constant - an integer between 100‘000‘000 and 200‘000‘000 inclusive. The second line should contain the output produced by the program if it were compiled and executed with the correct value of the constant.

Example

Output:

```
123456789
ARE YOU SOLVING IPSC PROBLEMS RIGHT NOW?
```

(syntactically valid (but incorrect) submission)

Added by: Roman Sol
Date: 2005-05-13
Time limit: 1s
Source limit: 1000B
Languages: All
Resource: IPSC 2005

SPOJ Problem Set (tutorial)

440. The Turtle´s Shortest Path

Problem code: TSHPATH

Given a list of cities. Each direct connection between two cities has its transportation cost (an integer bigger than 0). The goal is to find the paths of minimum cost between pairs of cities. Assume that the cost of each path (which is the sum of costs of all direct connections belongning to this path) is at most 200000. The name of a city is a string containing characters a,...,z and is at most 10 characters long.

Input

```
s [the number of tests <= 10]
n [the number of cities <= 10000]
NAME [city name]
p [the number of neighbours of city NAME]
nr cost [nr - index of a city connected to NAME (the index of the first city is 1)]
           [cost - the transportation cost]
r [the number of paths to find <= 100]
NAME1 NAME2 [NAME1 - source, NAME2 - destination]
[empty line separating the tests]
```

Output

```
cost [the minimum transportation cost from city NAME1 to city NAME2 (one per line)]
```

Example

```
Input:
1
4
gdansk
2
2 1
3 3
bydgoszcz
3
1 1
3 1
4 4
torun
3
1 3
2 1
4 1
warszawa
2
2 4
3 1
2
gdansk warszawa
bydgoszcz warszawa
```

```
Output:
3
2
```

Warning: large Input/Output data, be careful with certain languages

---

Added by:	Michał Małafiejski
Date:	2004-10-21
Time limit:	25s
Source limit:	50000B
Languages:	All
Resource:	DASM Programming League 2003 (thanks to Darek Dereniowski) a copy of SHPATH problem with 60s time limit

SPOJ Problem Set (tutorial)

450. Enormous Input Test

Problem code: INTEST

The purpose of this problem is to verify whether the method you are using to read input data is sufficiently fast to handle problems branded with the **enormous Input/Output** warning. You are expected to be able to process at least 2.5MB of input data per second at runtime.

Input

The input begins with two positive integers n k (n, k<=10^7). The next n lines of input contain one positive integer t_i, not greater than 10^9, each.

Output

Write a single integer to output, denoting how many integers t_i are divisible by k.

Example

Input:
```
7 3
1
51
966369
7
9
999996
11
```

Output:
```
4
```

Added by: Adrian Kosowski
Date: 2004-11-09
Time limit: 8s
Source limit:50000B
Languages: All
Resource: Idea put forward by Michael Mendelsohn

SPOJ Problem Set (tutorial)

453. Sums in a Triangle (tutorial)

Problem code: SUMTRIAN

This is problem `SUMITR` without strict source limit.

Let us consider a triangle of numbers in which a number appears in the first line, two numbers appear in the second line etc. Develop a program which will compute the largest of the sums of numbers that appear on the paths starting from the top towards the base, so that:

- on each path the next number is located on the row below, more precisely either directly below or below and one place to the right;
- the number of rows is strictly positive, but less than 100;
- all numbers are positive integers between O and 99.

Input

In the first line integer n - the number of test cases (equal to about 1000). Then n test cases follow. Each test case starts with the number of lines which is followed by their content.

Output

For each test case write the determined value in a separate line.

Example

```
Input:
2
3
1
2 1
1 2 3
4
1
1 2
4 1 2
2 3 1 1

Output:
5
9
```

Warning: large Input/Output data, be careful with certain languages

Added by:	Łukasz Kuszner
Date:	2004-11-10
Time limit:	2s
Source limit:	5000B
Languages:	All
Resource:	6-th International Olympiad In Informatics July 3-10. 1994. Stockholm - Sweden, Problem 1

SPOJ Problem Set (tutorial)

484. Fossil in the Ice

Problem code: TFOSS

A small group of archaeologists is working in the Antarctic. Their sensors have detected a number of caves in which there are interesting fossils. However, a thick layer of ice blocks the entrance to each cave. The archaeologists possess the equipment needed to burn a tunnel in the layer of ice, but the fuel is extremely expensive. In order to determine the size of each fossil the group has launched a number of probes through small bore-holes. Each probe which hit the fossil emits a signal consisting of its x and y coordinates. Your task is to determine the smallest possible size of the tunnel, which is equal to the maximal distance between any two probes (so that the fossil won't be damaged during extraction). The drilling equipment needs to be provided with the squared value of this distance.

Given the list of coordinates of the points containing probes, find the square of the maximal distance between any two probes.

Input

```
t [the number of tests <= 20]
[empty line]
n [the number of active probes <= 100000]
x1 y1 [coordinates of the first probe]
...
xn xn
[integer coordinates from -50000000 to 50000000]
[empty line]
[input for the next test cases...]
```

Text grouped in [ ] does not appear in the input file.

Output

```
o1 [the square of the maximal distance in the first set]
[output for the next test cases...]
```

Example

```
Input:
5

1
2 -3

4
0 0
-2 2
2 2
1 0

6
-4 2
```

```
2 2
5 0
0 5
6 1
-1 -1

10
0 0
5 1
9 2
12 3
14 4
15 5
16 7
17 10
18 14
19 19

10
2 -3
-1 2
0 5
-5 -1
-4 2
4 0
1 3
4 3
-3 -4
0 -2
```

Output:

```
0
16
101
722
98
```

Added by: Lukasz Wrona
Date: 2004-12-29
Time limit: 3s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

490. Armies

Problem code: ARMIES

Two enemy countries - *Bajtocja* and *Megabajtolandia* - are preparing for crucial war with each other. Each country has built an army consisting of some number of divisions, and each division consists of some number of soldiers. The way of waging the war, given by strategists from each contry, consists of sending the division with the most man power to fight, i.e. starting from the most numerous division to the least.

Thus, first each country will send its division with the most man power. If one of these divisions has more soldiers than the other, then the war is over and the winner is the owner of the larger division. If the man power of each of the divisions sent is the same then all the soldiers will kill each other and the next most numerous division is sent to fight. The man powers of the second divisions decide the war if and only if they are not the same. If not, the battle is carried on in aforementioned way. If, at some moment, one army runs out of divisions and the second one does not, then the war is over and the first army is the loser. If both armies run out of divisions then the war is over and there is a draw.
Give the result of the war, without any blood and murder.
Write a program, which:

- reads from standard input the description of *Bajtocja's* and *Megabajtolandia's* army, respectively,
- computes the result of the war,
- writes it to standard output.

Input

The first line of input contains one integer D ($1 <= D <= 30$) meaning the number of test cases. The description of each test case contains *4* lines. In the first, there is one integer B ($1 <= B <= 50\ 000$) meaning the number of divisions in *Bajtocja's* army. The second line contains B integers b_i ($1 <= b_i <= 1\ 000\ 000\ 000$) (separated by single space) meaning the man power (the number of soldiers) of consequtive divisions of *Megabajtolandia's* army. In the third line, there is one integer M ($1 <= M <= 50\ 000$) meaning the number of divisions of *Megabajtolandia's* army. The fourth line contains M integers m_i ($1 <= m_i < = 1\ 000\ 000\ 000$) (separated by single space) meaning the man power of consequtive divisions of *Megabajtolandia's* army.

Output

For each test case, your program should write, in separate lines, exactly one word:

- "Bajtocja" in case the winner is *Bajtocja*,
- "Megabajtolandia" in case the winner is *Megabajtolandia*,
- "Draw" in case of a draw.

Example

Sample input:
```
3
3
1 3 4
3
4 4 1
4
2 5 3 4
3
5 6 4
4
6 1 2 5
4
5 2 6 1
```

Sample output:
```
Megabajtolandia
Megabajtolandia
Draw
```

Added by:	Rafał Nowak
Date:	2005-02-07
Time limit:	3s
Source limit:	5000B
Languages:	All
Resource:	Winter sparing in Poznan, Poznan 2005 (22th January)

SPOJ Problem Set (tutorial)

491. The Cursed Room

Problem code: MMATCH

There is a school trip being organized for kids. The hotel the group is staying in can offer them one big room with enough beds to suit any group of visitors, and several smaller rooms with B beds alltogether. The children have heard many strange and frightening stories about the big room. That's why not even one of them wants to sleep in the big room. Furthermore not every kid would like to sleep in any bed.
Your goal is to assign B beds from the smaller rooms in such a way that the maximal number of children are happy (a child is happy when it gets to sleep in one of the beds it has selected).

Input

The first line contains a positive integer t<=1000 indicating the number of test cases. Each test case is an instance of the problem defined above. The first line of each test case is a pair of positive integers L and B (the number of children L<=100 and beds B<=100). The next lines contain a sequence of (c,b) pairs ending with two zeros. (c,b) means that the child c will be happy if it gets to sleep in bed b.

Output

For each test case print the maximal number of happy children.

Example

```
Input:
3
3 3
1 1
2 1
2 2
3 2
3 3
0 0
4 3
1 1
1 3
2 1
3 1
3 2
4 2
0 0
4 2
1 1
1 2
2 1
2 2
3 1
3 2
4 1
4 2
```

```
0 0
```

Output:

```
3
3
2
```

Added by: Tomasz Niedzwiecki
Date: 2005-01-18
Time limit: 5s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

500. Turbo Sort

Problem code: TSORT

Given the list of numbers, you are to sort them in non decreasing order.

Input

t - the number of numbers in list, then *t* lines follow [*t* <= 10^6].
Each line contains one integer: *N* [0 <= *N* <= 10^6]

Output

Output given numbers in non decreasing order.

Example

Input:

```
5
5
3
6
7
1
```

Output:

```
1
3
5
6
7
```

Added by:	Roman Sol
Date:	2005-03-14
Time limit:	5s
Source limit:	50000B
Languages:	All except: ERL JS
Resource:	ZCon

SPOJ Problem Set (tutorial)

503. Prime Intervals

Problem code: PRINT

In this problem you have to print all primes from given interval.

Input

t - the number of test cases, then *t* lines follows. [*t* <= 150]
On each line are written two integers *L* and *U* separated by a blank. *L* - lower bound of interval, *U* - upper bound of interval. [2 <= *L* < *U* <= 2147483647] [*U*-*L* <= 1000000].

Output

For each test case output must contain all primes from interval *[L; U]* in increasing order.

Example

Input:

```
2
2 10
3 7
```

Output:

```
2
3
5
7
3
5
7
```

Added by: Roman Sol
Date: 2005-03-28
Time limit: 9s
Source limit:15000B
Languages: All
Resource: ZCon

SPOJ Problem Set (tutorial)

511. Easy Sorting

Problem code: LEXISORT

Given is a list of words and a lexicographical ordering according to the ascii alphabet. Your task is to sort the words in increasing order.

Input

The first line contains the numbers of testcases k (k< 100). Every testcase c onsists of n+1 (1< n< 50000) lines. Each line contains of a string of 10 character s. The first line of each testcase contains n.

Output

Output the sorted list of words.

Example

```
Input:
2
2
helloworld
worldhello
2
aaaaaaaaaa
Aaaaaaaaaa
Output:
helloworld
worldhello
Aaaaaaaaaa
aaaaaaaaaa
```

Added by: Simon Gog
Date: 2005-04-13
Time limit: 4s
Source limit:8083B
Languages: All

SPOJ Problem Set (tutorial)

527. Just for Fun (Easy)

Problem code: J4FUN

birds
Puzzle ID: birds
Ten birds sit on a clothes line. We shoot and kill one of them. How many birds remain on the clothes line?
The answer for this puzzle consists of two lines, containing respectively:
- the ID of this puzzle
- one number: the number of birds that remain on the clothes line

bus
Puzzle ID: bus
A bus was travelling with less than 100 passengers. At stop A, exactly three quarters of the passengers got off and 7 passengers got on the bus. The same thing happened at next two stops, B and C. How many people got off at the stop C?
The answer for this puzzle consists of two lines, containing respectively:
- the ID of this puzzle
- the number of people getting off at C

palindrome
Puzzle ID: palindrome
Suppose we write dates in the MMDDYYYY format. In this format, the 2nd of October 2001 is a palindrome (a string equal to its reverse): 10022001. Find the previous date that yields a palindrome in this format.
The answer for this puzzle consists of two lines, containing respectively:
- the ID of this puzzle
- the 8-digit string

cube
Puzzle ID: cube
You have a cube NxNxN. How many straight cuts are necessary to cut it into N^3 cubes of size 1x1x1? You may arrange the pieces in any way you like before making each cut.
a) Solve for N=3
b) Solve for N=4
The answer for this puzzle consists of three lines, containing respectively:
- the ID of this puzzle
- the number of cuts from part a)
- the number of cuts from part b)

girl1
Puzzle ID: girl1
In a two-child family, one child is a boy.
What is the probability that the other child is a girl?
The answer for this puzzle consists of two lines, containing respectively:
- the ID of this puzzle
- the answer in the form a/b (where a,b are relatively prime)

girl2
Puzzle ID: girl2

In an unnamed overpopulated country the rulers agreed on a new law: Each woman may have as many children as she wants to, until she gives birth to a girl. After that, she may have no more children. Assume that the law will never be broken. All families will have as many children as they are (physically and legally) able to. On each birth either one boy or one girl is born with equal chances. In the current population the ratio males:females is 1:1. What will happen in the next 100 years?
A) The ratio of males to females will go up
B) The ratio of males to females will stay the same
C) The ratio of males to females will go down
The answer for this puzzle consists of two lines, containing respectively:
- the ID of this puzzle
- the uppercase letter corresponding to the correct answer

statements

Puzzle ID: statements
Given is a list with 2004 statements:
1. Exactly one statement on this list is false.
2. Exactly two statements on this list are false.
3. Exactly three statements on this list are false.
...
2004. Exactly 2004 statements on this list are false.
a) Determine which statements are true.
b) Replace "exactly" by "at least". Again, determine which statements are true.
The answer for this puzzle consists of three lines, containing respectively:
- the ID of this puzzle
- the encoded answer from part a)
- the encoded answer from part b)
How to encode the answer? If no statements are true, write the word 'NONE' (without the quotes). Otherwise take the set of true statements and write it as a set of ranges. E.g. the set {1,2,3,7,9,100,101} is encoded as 1-3,7,9,100-101

letters

Puzzle ID: letters
How many letters does the _shortest_ correct answer to this puzzle contain?
The answer for this puzzle consists of two lines, containing respectively:
- the ID of this puzzle
- your exact answer

century

Puzzle ID: century
The twentieth century ended on 31. 12. 2000, which was a Sunday. Looking into the future, on which days of the week won't any century ever end?
Remember that leap years are those divisible by 400 plus those divisible by 4 but not by 100. (1996 was a leap year, so was 2000, but 2100 won't be a leap year and neither will 2047.)
The answer for this puzzle consists of two lines, containing respectively:
- the ID of this puzzle
- the days of the week on which no century will ever end
The exact form of the answer is a comma-separated list of three-letter abbreviations of the days in the order in which they appear in a week. E.g. if the answer were Monday, Tuesday and Wednesday, write the string 'Mon,Tue,Wed' (without the quotes).

Input

There is no input for given problem.

Output

Output answers for each puzzle described below in the order they was described.

Example

Output:
```
birds
100
bus
10000
...
```

Added by: Roman Sol
Date: 2005-05-18
Time limit: 1s
Source limit:10000B
Languages: C99 strict TEXT
Resource: IPSC 2005

SPOJ Problem Set (tutorial)

536. How many Fibs

Problem code: TFIB

Recall the definition of the Fibonacci numbers:

f_1 := 1
f_2 := 2
$f_n := f_{n-1} + f_{n-2}$ (n>=3)

Given two numbers *a* and *b*, calculate how many Fibonacci numbers are in the range [*a*,*b*].

Input

The input contains several test cases. Each test case consists of two non-negative integer numbers *a* and *b*. Input is terminated by *a=b=0*. Otherwise, $a<=b<=10^{100}$. The numbers *a* and *b* are given with no superfluous leading zeros.

Output

For each test case output on a single line the number of Fibonacci numbers f_i with $a<=f_i<=b$.

Example

Input:

```
10 100
1234567890 9876543210
0 0
```

Output:

```
5
4
```

Added by:	Adrian Kuegel
Date:	2005-07-05
Time limit:	10s
Source limit:	50000B
Languages:	All
Resource:	University of Ulm Local Contest 2000

SPOJ Problem Set (tutorial)

644. Byteland Money Exchange

Problem code: PAYBACK

In Byteland after the food shortage banks made credit payment freely available. At the end of the year companies have to settle their debts and to give a statement on their cashflow to the Claim Office. Among Banks and Companies a net of debts was created. Unfortunetely Banks gave a sky-high price on money transfers. For company owners it was unprofitable to pay all money transfers as they were. They chose you to help them out. Your assignment is to balance the debt network.
>
> You are given **[t<=1000]** test cases- a test case consists of the size **[N<=1000]** of the debt network, followed by a description of the network itself. Each line consists of integers separated by spaces ending with a new line. Each value states how much money the company in line "i" is in debt to company "j" where "j" is the column number.
>
> Your assignment is to limit the number of money transfers by determining which companies are in debt, which have earned money and which have shown neither profit nor loss.
>
>

Input

t [- test cases]
> N [- size of the debt net]
> a[1,1] a[1,2] a[1,3] ... a[1,n]
> a[2,1] a[2,2] a[2,3] ... a[2,n]
> ...[[debt size for each company - a[1,3] denotes the sum borrowed by company 1 from 3]]
> ...
> a[n,1] a[n,2] a[n,3] ... a[n,n]
> [empty line]
> [next test case]
>
> a graphical example
>
> INPUT Graph
>
>

Output

T[Size of solution]
> a[1,1] a[1,2] a[1,3] ... a[1,n]
> a[2,1] a[2,2] a[2,3] ... a[2,n]
> ...[[debt size for each company - a[1,3] denotes the sum borrowed by company 1 from 3]]
> ...
> a[n,1] a[n,2] a[n,3] ... a[n,n]

> [empty line]
> [next solution]
>

> [All the output data should be integers.]
> Text grouped in [ ] does not appear in the input file.
>
> a graphical example
>
> OUTPUT Graph1
>
> a graphical example- same input
>
> OUTPUT Graph2
>
>

Example

Input:

```
1
7
0 18 25 34 14 21 40
44 0 64 0 11 5 24
11 35 0 23 17 26 23
19 50 20 0 16 7 0
0 14 9 0 0 27 18
42 5 17 8 3 0 17
36 26 0 47 7 6 0
```

Output:

```
7
0 10 0 0 0 0 0
0 0 0 0 0 0 0
0 12 0 0 0 6 3
0 0 0 0 0 0 59
0 0 0 0 0 0 29
0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

Added by: Sylwester Herber
Date: 2005-12-01
Time limit: 10s
Source limit:50000B
Languages: All
Resource: inspired by "Algorithm Complexity Theory" project assignment 2002

SPOJ Problem Set (tutorial)

691. Hotel Floors

Problem code: HFLOOR

We are given a top view of a hotel floor, which is represented by an MxN matrix of characters, composed of (only) the following:

'#' is a Wall
'-' is Free Space
'*' is an occupied space (by a single person).

We are required to evaluate the average number of people living in a room.

Constraints:
M, N <= 100
Number of test-cases<= 10
All border edges of the map will be walls.
There will be at least one room.

Input

The 1st line contains the number of test inputs, with each test case consisting of:
M N
MxN matrix of characters

Output

For each test case output a line with the average number of people living per room, rounded to *exactly* two decimal places.

Example

```
Input:
2
5 5
#####
#**##
###*#
#**##
#####
6 10
##########
#---*--*##
###-*----#
#**#######
##***---##
```

```
##########
```

Output:

```
1.67
4.00
```

Added by:	Prasanna
Date:	2006-01-12
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	ByteCode ’06

SPOJ Problem Set (tutorial)

692. Fruit Farm

Problem code: FFARM

We visited a farm, which was barren except for certain points in which fruit trees existed. In general it was true that only places with palindromic indices contained fruit trees. We are required to buy a subregion of this farm of length at most L so that our aims (in the given priority) are satisfied best.

1. Maximize the amount of fruit trees present in it.
2. Minimize the size (length) of the farm bought.
3. Select the farm whose beginning is leftmost.

Input

The 1st line contains the number of test cases, T<=20, each test case:
A B L
where [A,B] is the closed interval of land which we visited.

Output

S E
where [S,E] is the closed interval of land which we buy.
If there is no fruit-tree in the visited interval, print "Barren Land."

Constraints:
1<=A<=B<=1000

Example

Input:
```
6
1 10 5
800 1000 5
80 120 5
30 60 12
12 18 40
23 30 10
```

Output:
```
1 5
808 808
99 101
33 44
Barren Land.
Barren Land.
```

Added by:	Prasanna
Date:	2006-01-13
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	ByteCode ’06

SPOJ Problem Set (tutorial)

732. Johnsons Algorithm

Problem code: JHNSN

Johnson's algorithm solves the all-pairs shortest path problem in a weighted, directed graph.

Input

```
t [number of test graphs]

[description of each graph]
n [number of nodes, 1<=n<=100]
m [number of edges, m>=0]
[next the list of edges, one edge per line]
u v w [e(u,v) - edge from node u to node v with weight w]
      [1<=u<=n, 1<=v<=n, -1000<=w<=1000]
...   [next edge]

[next graph]
...
```

Output

If the *i*-th test graph has negative weight cycles, then the answer should be:

```
graph i no [where 'i' is the number of the graph, 1<=i<=t]
```

Otherwise you should output the following data:

```
graph i yes

[vector of function h(v)]
h 1  h 2  ... h n+1

[matrix d[u,v], the solution of the all-pairs shortest path problem]
d 1,1  d 1,2  ... d 1,n
d 2,1  d 2,2  ... d 2,n
... ... ... ...
d n,1  d n,2  ... d n,n
[if the path doesn't exist, you should output # instead]
```

Example

Input:

```
6

2
2
1 2 -2
2 1 1

6
```

```
8
1 2 8
1 6 6
6 2 3
2 3 -1
3 6 -2
6 5 -2
5 4 2
3 4 3

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

2
0

1
0

2
2
1 2 -1
2 1 0
```

Output:

```
graph 1 no

graph 2 yes

0 0 -1 -3 -5 -3 0

0 8 7 5 3 5
# 0 -1 -3 -5 -3
# 1 0 -2 -4 -2
# # # 0 # #
# # # 2 0 #
# 3 2 0 -2 0

graph 3 yes

0 0 0 0 0

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

graph 4 yes

0 0 0

0 #
# 0

graph 5 yes

0 0
```

```
0

graph 6 no
```

Added by: Bartłomiej Kowalski
Date: 2006-02-05
Time limit: 10s
Source limit:50000B
Languages: All
Resource: http://www.sphere.pl/~deren/gms/03_najkrsc2.pdf

SPOJ Problem Set (tutorial)

762. Problems Collection (Volume 1)

Problem code: PCV1

Problem 1 It is easily proved that no equilateral triangle exists with integral length sides and integral area. However, the *almost equilateral triangle* 5-5-6 has an area of 12 square units. We shall define an *almost equilateral triangle* to be a triangle for which two sides are equal and the third differs by no more than one unit. Find the sum of the perimeters of every *almost equilateral triangles* with integral side lengths and area and whose perimeters do not exceed one billion (1,000,000,000).
Problem 2 If a box contains twenty-one coloured discs, composed of fifteen blue discs and six red discs, and two discs were taken at random, it can be seen that the probability of taking two blue discs, P(BB) = (15/21)*(14/20) = 1/2. The next such arrangement, for which there is exactly 50% chance of taking two blue discs at random, is a box containing eighty-five blue discs and thirty-five red discs. By finding the first arrangement to contain over 10^12 = 1,000,000,000,000 discs in total, determine the number of blue discs that the box would contain.
Problem 3 Consider the fraction, n/d, where n and d are positive integers. If n < d and HCF(n,d)=1, it is called a reduced proper fraction. If we list the set of reduced proper fractions for d <= 8 in ascending order of size, we get: 1/8, 1/7, 1/6, 1/5, 1/4, 2/7, 1/3, **3/8**, **2/5**, **3/7**, 1/2, 4/7, 3/5, 5/8, 2/3, 5/7, 3/4, 4/5, 5/6, 6/7, 7/8 It can be seen that there are 3 fraction between 1/3 and 1/2. How many fractions lie between 1/3 and 1/2 in the sorted set of reduced proper fractions for d <= 10,000?
Problem 4 The series, 1^1 + 2^2 + 3^3+ ... + 10^10 = 10405071317. Find the last ten digits of the series, 1^1 + 2^2 + 3^3+ ... + 1000^1000
Problem 5 The cube, 41063625 (345^3), can be permuted to produce two other cubes: 56623104 (384^3) and 66430125 (405^3). In fact, 41063625 is the smallest cube which has exactly three permutations of its digits which are also cube. Find the smallest cube for which exactly five permutations of its digits are cube.
Problem 6 Euler’s Totient function, phi(n), is used to determine the number of numbers less than n which are relatively prime to n. For example, as 1, 2, 4, 5, 7, and 8, are all less than nine and relatively prime to nine, phi(9) = 6. Interestingly, phi(87109) = 79180,and it can be seen that 87109 is a permutation of 79180. Find the value of n, below ten million, for which phi(n) is a permutation of n and the ratio n/phi(n) - produces a minimum.
Problem 7 The prime 41, can be written as the sum of six consecutive primes: 41 = 2 + 3 + 5 + 7 + 11 + 13 This is the longest sum of consecutive primes that adds to a prime below one-hundred. The longest sum of consecutive primes below one-thousand that adds to a prime, contains 21 terms, and is equal to 953. Which prime, below one-million, can be written as the sum of the most consecutive primes? (Integer 1 isn’t prime)
Problem 8 145 is a curious number, as 1! + 4! + 5! = 1 + 24 + 120 = 145. Find the sum of all numbers which are equal to the sum of the factorial of their digits. Note: as 1! = 1 and 2! = 2 are not sums they are not included.
Problem 9 A permutaion is an ordered arrangement of objects. For example, 3124 is one possible permutation of the digits 1, 2, 3 and 4. If all of the permutations are listed numerically or alphabetically, we call it lexicographic order. The lexicographic permutations of 0, 1 and 2 are: 012, 021, 102, 120, 201, 210 What is the millionth lexicographic permutation of the digits 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9?
Problem 10 By counting carefully it can be seen that a rectangular grid measuring 3 by 2 contains eighteen rectangles. Although there exists no rectangular grid that contains exactly two million

rectangles, find the area of the grid with the nearest solution.

Input

There is no input for this problem.

Output

Output answer as the set of lines. On each line first is number of problem and second is answer for this problem. If any of answers will be incorrect, you'll recieve Wrong Answer.

Score

For each solved problem you'll recieve exactly one point (10 points maximum, if all problems are solved correctly).

Example

Output:

```
1 6174046
2 6568127473
5 806257
8 51146700
```

It's just the example how output would look like. If all 4 answers correct (1, 2, 5 and 8 problems), you'll recieve 4 points.

Added by: Roman Sol
Date: 2006-01-23
Time limit: 1s
Source limit:5000B
Languages: TEXT
Resource: ZCon 2006

SPOJ Problem Set (tutorial)

1023. Arranging Dominoes

Problem code: ADOMINO

Dominoes have long entertained both game enthusiasts and programmers for quite some time. Many games can be played with dominoes, including multiplayer and single player games. Hari Khan has come up with a single player game. He takes **N** boxes and arranges them in a row at positions $\mathbf{N_1}$, $\mathbf{N_2}$ ... $\mathbf{N_N}$. Now he has to place **D** dominoes (**D** <= **N**) in the boxes such that the minimum distance between any two filled boxes is maximized.

Input

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

The first line of each test case consists of two integers, **N** <= 100000 and **D** <= **N**, separated by a single space.

N lines follow, each containing a single integer $\mathbf{N_i}$ <= 1000000000, indicating the location of the i^{th} box.

Output

For each test case, output a single line containing a single integer denoting the largest minimum distance achievable between any two boxes with dominoes.

Example

```
Input:
1
5 3
1
2
3
4
5

Output:
2
```

Added by: Matthew Reeder
Date: 2006-10-29
Time limit: 3s
Source limit:30000B
Languages: All
Resource: Al-Khawarizm 2006

SPOJ Problem Set (tutorial)

1118. Snowman

Problem code: SNOWMAN

Finally the time of the year has come where children can build snowmans. The children have collected some snow and ask you how big the snowman is going to be.

Assume that the snowman will consist of three spheres stacked on top of each other. The lower two spheres are of equal size, the smaller sphere (used for the head) will have a radius of 25 percent of the radius of the larger spheres.

Input

The first line of the input contains a number **t** <= 100, which indicates the number of test cases to follow. Each test case consists of a line with one integer **a** <= 500000, the amount of snow in cm^3.

Output

For each test case, print a line with the height of the snowman in cm. Round this number down to the next smaller integer.

Example

```
Input:
2
100
500000

Output:
10
175
```

Added by: Adrian Kuegel
Date: 2006-11-28
Time limit: 5s
Source limit: 5000B
Languages: All

SPOJ Problem Set (contest)

1238. Special Nim Game

Problem code: NIMGAME

In this variant of the Nim game, a pile of **N** stones is placed between two players. The players take alternating turns and remove some stones. The player who takes the last stone wins.

There are two restrictions however:

1. The first player has to remove between 1 and **N**-1 stones.
2. After the first move, the next player has to remove between 1 and 2.**k** stones, where **k** is the number of stones removed in the last move.

If both players play perfectly, then it is possible to determine which player will win the game. Note that during the game the game state can be described by the number of remaining stones and the number of stones which can be taken in the next move. Each game state is either a winning position or a losing position.

You have to determine for which values of **N** (2 <= **N** <= 2000) the second player has a winning strategy.

Input

There is no input for this problem.

Output

Print the values **N** for which the second player has a winning strategy.

Example

```
Output:
2
3
5
...
1597
```

Obviously, the example output is incomplete and shows only the first three values and the last value to be printed.

Added by: Adrian Kuegel
Date: 2007-01-18
Time limit: 5s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

1239. Recurrence Equation Finder

Problem code: RECEQU

Many problems have solutions involving linear recurrence equations of the form **f(n) = a . f(n-1) + b . f(n-2) (n >= 2)**. Usually the coefficients **a** and **b** are between 0 and 10, so it would be useful to have a program which checks if some given values can be produced by such a recurrence equation. Since the growth of the values **f(n)** can be exponential, we will consider the values modulo some integer constant **k**.

More specifically you will be given **f(0)**, **f(1)**, **k** and some value pairs (**i** , $\mathbf{x_i}$), where $\mathbf{x_i}$ is the remainder of the division of **f(i)** by **k**.

You have to determine coefficients **a** and **b** for the recurrence equation **f** such that for each given value pair (**i**, $\mathbf{x_i}$) the equation $\mathbf{x_i}$ **= f(i) mod k** holds.

Hints

You can write the recurrence equation as follows:

Let [IMAGE] n , the identity $A^n = (A^{n \text{ div } 2})^2 \cdot A^{n \bmod 2}$ may be used. Also, (a . b) mod c = ((a mod c) . (b mod c)) mod c.

Input

The first line of the input contains a number **T <= 20** which indicates the number of test cases to follow.
Each test case consists of 3 lines.
The first line of each test case contains the three integers **f(0)**, **f(1)** and **k**, where **2 <= k <= 10000** and **0 <= f(0),f(1) < k**

Output

For each test case print one line containing the values **a** and **b** separated by a space character, where **0 <= a,b <= 10**.
You may assume that there is always a unique solution.

Example

Input:

```
2
1 1 1000
3
2 2 3 3 16 597
0 1 10000
4
11 1024 3 4 1000000000 4688 5 16
```

Output:

```
1 1
2 0
```

Added by: Adrian Kuegel
Date: 2007-01-18
Time limit: 10s
Source limit:50000B
Languages: All

SPOJ Problem Set (contest)

1279. Run Length Decoding

Problem code: RLDEC

Your task is to write a program that decodes a sequence of characters which was encoded using a simple form of run-length encoding, as described by the rules below.

Any sequence of between 2 to 9 identical characters is encoded by two characters. The first character is the length of the sequence, represented by one of the characters 2 through 9. The second character is the value of the repeated character. A sequence of more than 9 identical characters is dealt with by first encoding 9 characters, then the remaining ones.

Any sequence of characters that does not contain consecutive repetitions of any characters is represented by a 1 character followed by the sequence of characters, terminated with another 1. If a 1 appears as part of the sequence, it is escaped with a 1, thus two 1 characters are output.

Input

The first line in the input contains a number **T <= 200** which specifies the number of test cases to follow. Each test case consists of one line with the encoding of a sequence of characters. Each line consists of letters (both upper- and lower-case), digits, spaces, and punctuation and is terminated with a newline character. No other characters appear in the input. You may assume that each line is a valid encoding of some sequence of characters.

Output

For each line in the input print one line with the decoded sequence of characters.

Example

Input:
```
3
9A1ABC131
1112 3124
111111
```

Output:
```
AAAAAAAAAABC111
12 344
11
```

Added by: Adrian Kuegel
Date: 2007-01-27
Time limit: 10s
Source limit:50000B
Languages: All

SPOJ Problem Set (contest)

1282. How many Islands

Problem code: COUNTISL

You are given a simple map of an archipelago. Can you determine how many islands it shows?

The map consists of grid squares with characters, where '#' indicates land and '.' indicates water. Two land squares belong to the same island if they are neighbouring grid squares, which means their x coordinates and y coordinates differ by at most 1.

Input

The first line of the input contains a number **T** <= 20 which indicates the number of test cases to follow.

Each test case starts with a line containing two numbers **n** and **m** (1 <= **n,m** <= 200), the number of rows and the number of columns of the grid, respectively. The following **n** lines contain **m** characters each and describe the map to be processed. You may assume that the map contains only characters '#' and '.', and that the border of the map consists only of water (character '.').

Output

For each test case print in a line the number of islands shown on the corresponding map in the input.

Example

```
Input:
2
1 1
.
6 4
....
..#.
.#..
....
.##.
....

Output:
0
2
```

Added by: Adrian Kuegel
Date: 2007-01-29
Time limit: 10s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

1392. Sum of Factors

Problem code: CZ_PROB2

Find the sum of the factors of a number including 1 and the given number.

Input

Number of test cases T followed by T lines of number n.
0 < T <= 5000
0 < n <= 999999999
Note: The number will not have a very large prime factor.

Output

The sum of the factors for each test case.

Example

```
Input:
2
6
5
Output:
12
6
```

Added by:	Rahul
Date:	2007-03-10
Time limit:	1s
Source limit:	4000B
Languages:	All
Resource:	Siddharth Agarwal

SPOJ Problem Set (tutorial)

1393. Ping Pong Probability

Problem code: CZ_PROB3

Two Ping Pong players agree to play several games. The players are evenly matched. However, the person serving first has a probability p of winning that game. A serves the first game and thereafter the loser serves first. What is the Probability that A wins the nth game?

Input

Number of test cases T followed by T lines of 'n' - the number of games played; and 'p' probability of person serving first winning.
$0 < T < 100$

Output

Print for each test case the probability of A winning.
Please print a *double* value.

Example

```
Input:
2
4 0.7
7 0.7

Output:
0.4872
0.500819
```

Added by: Rahul
Date: 2007-03-10
Time limit: 1s
Source limit:5000B
Languages: All
Resource: Dilip Rajeev

SPOJ Problem Set (tutorial)

1394. Dividing Spaces

Problem code: CZ_PROB4

Into how many parts can k cuts using an n-1 dimensional hyper-plane divide an n- dimensional hypercube?
Hint: Experiment with n=2, n=3 find a pattern
Eg: n=3, k=5 => dividing a cube with 5 cuts using planes.

Input

T, number of test cases followed by T lines of 'n' and 'k'.

Output

The number of parts in separate lines for each of 'T' test cases.

Example

```
Input:
2
14 20
6 23

Output:
1026876
145499
```

Added by: Rahul
Date: 2007-03-11
Time limit: 10s
Source limit:50000B
Languages: All
Resource: Dilip Rajeev

SPOJ Problem Set (tutorial)

1397. Put Them on a Circle

Problem code: CZ_PROB7

An array of numbers (N1, N2, N3... Nn) is given. The numbers are to be placed on a circle such that the sum of any two adjacent numbers is not divisible by a number in the set of Numbers V1, V2 ,V3...Vk . Write a function, that given N and V determines if such an arrangement exists.

Input

T, the number of test-cases.
For each test case: Input array size of N, ’n’ and array size of V, ’k’. This is followed by one line of values of array N (separated by spaces) and then one line of values of array ’V’(separated by spaces).

Output

For each test case print "yes" or "no" on a separate line.

Example

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

Output:
yes
no
```

Added by: Rahul
Date: 2007-03-11
Time limit: 3s
Source limit:7000B
Languages: All
Resource: Dilip Rajeev

SPOJ Problem Set (main)

1415. Problems Collection (Volume 2)

Problem code: PCV2

Problem 1 How many consecutive positive integers can you find, such that the sum of digits (in decimal representation) of each of them is not divisible by 13?
Note: Because 49 is the first number for which the sum of digits divisible by 13, so for instance integers from 1 to 48 satisfy the condition.
Problem 2 You can find the solution in this file: answer.zip
Problem 3 Find the answer in this picture:

Find the Answer

Problem 4 When looking at a number from left-to-right, if no digit is smaller than the digit to its left, then the number is called *increasing*; for example, 125589.
Similarly if no digit is smaller than the digit to its right, the number is called *decreasing*; for example, 995421.
We shall call a positive integer that is neither increasing nor decreasing a "bouncy" number; for example, 64783.
Clearly there cannot be any bouncy numbers below one-hundred, but just over half of the numbers below one-thousand (525) are bouncy. In fact, the smallest number for which the proportion of bouncy numbers first exceeds 50% is 538. Bouncy number become more and more common and by the time we reach 21780 the proportion of bouncy numbers is equal to 90%.
Find the least number for which the proportion of bouncy numbers is exactly 99%.
Problem 5 The radical of n, rad(n) - is the product of distinct prime factors of n. For example, 1008 = 2^4*3^2*7 , so rad(1008) = 2*3*7 = 42.
If we calculate rad(n) for 1 <= n <= 10, then sort them with respect to rad(n), breaking ties by sorting with respect to the value of n, we get:
Unsorted:
n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
rad(n) = 1, 2, 3, 2, 5, 6, 7, 2, 3, 10
Sorted:
n = 1, 2, 4, 8, 3, 9, 5, 6, 7, 10
rad(n) = 1, 2, 2, 2, 3, 3, 5, 6, 7, 10
k = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
Let E(k) be the kth element in the sorted n column; for example, E(4) = 8 and E(6) = 9.
If rad(n) is sorted for 1 <= n <= 100000, find E(10000).
Problem 6 Consider the infinite polynomial series: AF(x) = x*F(1) + x^2*F(2) + x^3*F(3) + ..., where F(k) is the kth term in the Fibonacci sequence: 1, 1, 2, 3, 5, 8, ..., that is, F(1) = 1; F(2) = 1; F(k) = F(k-1) + F(k-2).
For this problem we shall be interested in values of x for which AF(x) is a positive integer.
Surprisingly AF(1/2) = 1/2 + 1/4 + 2/8 + 3/16 + 5/32 + ... = 2
The corresponding values of x for the first five natural numbers are shown below:
1) x = sqrt(2)-1 AF(x) = 1
2) x = 1/2 AF(x) = 2
3) x = (sqrt(13)-2)/3 AF(x) = 3
4) x = (sqrt(89)-5)/8 AF(x) = 4

5) x = (sqrt(34)-3)/5 AF(x) = 5
We shall call AF(x) a golden nugget if x is rational, because they become increasingly rarer; for example, the 10th golden nugget is 74049690.
Find the 15th golden nugget.
Problem 7 Euler's totient function F(n) is defined as the number of positive integers not exceeding n that are relatively prime to n, where 1 is counted as being relatively prime to all numbers. So, for example, F(20) = 8, because the eight integers 1, 3, 7, 9, 11, 13, 17, and 19 are relatively prime to 20. The table below shows values of F(n) for the first 10 integers:
F(1) = 1, F(2) = 1, F(3) = 2, F(4) = 2, F(5) = 4,
F(6) = 2, F(7) = 6, F(8) = 4, F(9) = 6, F(10) = 4
Euler's totient valence function v(n) is defined as the number of positive integers k such that F(k) = n. For instance, v(8) = 5 because only the five integers k = 15, 16, 20, 24, and 30 are such that F(k) = 8. The table below shows values of v(n) for n <= 16. (For n not in the table, v(n) = 0).

n	v(n)	k such that F(k)=n
1	2	1, 2
2	3	3, 4, 6
4	4	5, 8, 10, 12
6	4	7, 9, 14, 18
8	5	15, 16, 20, 24, 30
10	2	11, 22
12	6	13, 21, 26, 28, 36, 42
16	6	17, 32, 34, 40, 48, 60

Evaluate v(2^1000).
Problem 8 In how many ways can 50! be expressed as a sum of two or more consecutive positive integers?
Problem 9 Imagine you have a crystal in the shape of an equilateral triangle, one unit long on each side. In the right conditions, the crystal starts to grow. After one minute, two sides grow from each side of the triangle that are perfectly symmetrical. The result is a six-pointed star, that has sides that are exactly 1/3 of the length of the side they grew from. After another minute, each side sprouts two more sides that are exactly 1/3 of the length of the side they came from. See the pictures to get a better idea:

[IMAGE] [IMAGE]
[IMAGE] [IMAGE]
[IMAGE] [IMAGE]
[IMAGE] [IMAGE]

Your challenge is to find the perimeter (rounded to the nearest whole number) after one hour, thirty-three minutes.
Problem 10 One more picture:

Number 758932

Input

There is no input for this problem.

Output

Output the answer as a set of lines. In each line first give the number of the problem and then the answer to this problem. If any of the answers are incorrect, you will receive Wrong Answer.

Score

For each solved problem you'll recieve exactly one point (10 points maximum, if all problems are solved correctly).

Example

Output:

```
1 6174046
2 AnsweR
5 806257
8 51146700
```

It's just the example of what the output should look like. If all 4 answers are correct (problems 1, 2, 5 and 8), you will recieve 4 points.

Added by:	Roman Sol
Date:	2006-04-11
Time limit:	1s
Source limit:	10000B
Languages:	TEXT
Resource:	ZCon 2007

SPOJ Problem Set (tutorial)

1428. Easy sudoku

Problem code: EASUDOKU

You are to solve the classic 9x9 sudoku problem.

Input

The first line contains only one number - number of test cases (more than 1 and less then 15). Than the test cases are given. Each of them is given by 81 numbers (from 0 to 9) seperated by single white space (new line after each 9th number). Zero means that it is to solve by your program.

Output

In case of there does not exist the solution write "No solution". If there exist 81 number beeing the solution of sudoku problem, you have to write all those 81 numbers (separated like in input).

Example

```
Input:
2
0 0 0 0 6 9 8 3 0
9 8 0 0 0 0 0 7 6
6 0 0 0 3 8 0 5 1
2 0 5 0 8 1 0 9 0
0 6 0 0 0 0 0 8 0
0 9 0 3 7 0 6 0 2
3 4 0 8 5 0 0 0 9
7 2 0 0 0 0 0 6 8
0 5 6 9 2 0 0 0 0

0 0 0 0 6 9 8 3 0
9 8 0 0 0 0 0 7 6
6 0 0 0 3 8 0 5 1
2 0 5 4 8 1 0 9 0
0 6 0 0 0 0 0 8 0
0 9 0 3 7 0 6 0 2
3 4 0 8 5 0 0 0 9
7 2 0 0 0 0 0 6 8
0 5 6 9 2 0 0 0 0

Output:
5 1 2 7 6 9 8 3 4
9 8 3 5 1 4 2 7 6
6 7 4 2 3 8 9 5 1
2 3 5 6 8 1 4 9 7
1 6 7 4 9 2 3 8 5
4 9 8 3 7 5 6 1 2
3 4 1 8 5 6 7 2 9
7 2 9 1 4 3 5 6 8
8 5 6 9 2 7 1 4 3
No solution
```

Added by:	Rafał Nowak
Date:	2007-03-23
Time limit:	1s
Source limit:	5000B
Languages:	All

SPOJ Problem Set (tutorial)

1474. Charge

Problem code: TREE3

Network is becoming more and more important in the modern times. There are hundreds million of people studying,researching and playing with the Internet. However, we can't forget that there will be a lot of cost when the network is running. So charging from the users is necessary and of course reasonable.

The very very famous Southern Moutain high School in the City of Soft Sheep has such a network of education. There are 2^N users in total, which are numbered 1,2,3...2^N . These users are connected by routers and cables.

Users,routers,cables make a Full Binary Tree together. Each leaf(colored white) of the tree denotes a user,each non-leaf node(colored gray) denotes a router,each edge denote a cable,see the following picture.

[IMAGE]

The charge mode of the network company in the city of Soft Sheep is quite strange, so called "Pairing Charging". It means that they charge from each two users i and j (1 <= i < j <= 2^N). Users can choose one mode of charge among A and B by themselves,so the cost that the company charge from the great school is relative to the mode of charging by each user. The total cost equals to the sum of the cost of each pair of users.

Some definations:

- **Ancestor**:The root of the tree has no Ancestor,each ancestor of some other node Ancestors are father of this node and the father's Ancestor.
- **dominated Leaf**:The leaves dominate nothing,the leaves dominated by one non-leaf node are all the leaves dominated by the left and right child of this node.
- **Dist**:The shortest path between each pair of nodes in the tree.

For each pair of users i,j(1 <= i < j <= 2^N),first we find the LCA(Least Common Ancestor) of the two node named P,then let's consider the Donimated Leaves of P(the users assign to P). We define nA,nB denoted the number of users choose A and B to charge in these Donimated Leaves.

Charging is following the rule below:(in the rule,F(i,j) denotes the flux between i and j and will be given.)

[IMAGE]

Since the total cost is relative to the mode of charging,the users in the great Southern Moutain School hope to minimize the cost by changing the way of charging.However,the company has recorded the mode that each user choosed when they registered.So for each user i,if he/she wants to change the mode of charging,(change from mpde A to mode B,or change from mode B to mode A),he/she must pay $Ci to the company to modify the record.

Your task is:

Given the mode the users chosen when they registered,and Ci,decide the mode to charge of each user to minimize the total cost(the cost of changing mode + the sum of the cost of the Pairing Charging).

Input

```
T [The number of test cases]
N [N<=10]
D1 D1 D2 ... DM [M=2^N, Di=0 iff the mode user i chosen when he/she registerd is A and Di=1 otherwise.]
C1 C1 C2 ... CM [the cost of changing the mode of each user,0<=Ci<=500000]
F(1,2) F(1,3) ... F(1,M)
F(2,3) F(2,4) ... F(2,M)
...
F(M-2,M-1) F(M-2,M)
F(M-1,M)
[The table above is the flux table description,0<=F(i,j)<=500]
[other tests]
```

Output

```
TheMinCost
[other tests]
```

Example

Sample Input:

```
1
2
1 0 1 0
2 2 10 9
10 1 2
2 1
3
```

Sample Output:

```
8
```

Hints:

```
Change the mode of the first user from mode B to mode A.
```

Added by: Blue Mary
Date: 2007-04-01
Time limit: 17s
Source limit:50000B
Languages: All except: C99 strict
Resource: Chinese National Olympiad in Informatics 2006,Day 1(co-author lcosvse)

SPOJ Problem Set (tutorial)

1679. Annoying painting tool

Problem code: ANNOYING

Maybe you wonder what an annoying painting tool is? First of all, the painting tool we speak of supports only black and white. Therefore, a picture consists of a rectangular area of pixels, which are either black or white. Second, there is only one operation how to change the colour of pixels:

Select a rectangular area of **r** rows and **c** columns of pixels, which is completely inside the picture. As a result of the operation, each pixel inside the selected rectangle changes its colour (from black to white, or from white to black).

Initially, all pixels are white. To create a picture, the operation described above can be applied several times. Can you paint a certain picture which you have in mind?

Input Specification

The input contains several test cases. Each test case starts with one line containing four integers **n**, **m**, **r** and **c**. (*1 <= r <= n <= 100, 1 <= c <= m <= 100*), The following **n** lines each describe one row of pixels of the painting you want to create. The **i** $^{\mathbf{th}}$ line consists of **m** characters describing the desired pixel values of the **i** $^{\mathbf{th}}$ row in the finished painting ('0' indicates white, '1' indicates black).

The last test case is followed by a line containing four zeros.

Output Specification

For each test case, print the minimum number of operations needed to create the painting, or -1 if it is impossible.

Sample Input

```
3 3 1 1
010
101
010
4 3 2 1
011
110
011
110
3 4 2 2
0110
0111
0000
0 0 0 0
```

Sample Output

```
4
6
-1
```

Added by:	Adrian Kuegel
Date:	2007-07-06
Time limit:	10s
Source limit:	50000B
Languages:	All
Resource:	University of Ulm Local Contest 2007

SPOJ Problem Set (tutorial)

1680. Black and white painting

Problem code: BLACK

You are visiting the Centre Pompidou which contains a lot of modern paintings. In particular you notice one painting which consists solely of black and white squares, arranged in rows and columns like in a chess board (no two adjacent squares have the same colour). By the way, the artist did not use the tool of problem A to create the painting.

Since you are bored, you wonder how many 8×8 chess boards are embedded within this painting. The bottom right corner of a chess board must always be white.

Input Specification

The input contains several test cases. Each test case consists of one line with three integers **n**, **m** and **c**. ($8 <= n, m <= 40000$), where **n** is the number of rows of the painting, and **m** is the number of columns of the painting. **c** is always *0* or *1*, where *0* indicates that the bottom right corner of the painting is black, and *1* indicates that this corner is white.

The last test case is followed by a line containing three zeros.

Output Specification

For each test case, print the number of chess boards embedded within the given painting.

Sample Input

```
8 8 0
8 8 1
9 9 1
40000 39999 0
0 0 0
```

Sample Output

```
0
1
2
799700028
```

Added by: Adrian Kuegel
Date: 2007-07-06
Time limit: 10s
Source limit:50000B
Languages: All
Resource: University of Ulm Local Contest 2007

SPOJ Problem Set (tutorial)

1682. Deli Deli

Problem code: DELI

Mrs. Deli is running the delicatessen store "Deli Deli". Last year Mrs. Deli has decided to expand her business and build up an online store. She has hired a programmer who has implemented the online store.

Recently some of her new online customers complained about the electronic bills. The programmer had forgotten to use the plural form in case that an item is purchased multiple times. Unfortunaly the programmer of Mrs. Deli is on holiday and now it is your task to implement this feature for Mrs. Deli. Here is a description how to make the plural form:

1. If the word is in the list of irregular words replace it with the given plural.
2. Else if the word ends in a consonant followed by "y", replace "y" with "ies".
3. Else if the word ends in "o", "s", "ch", "sh" or "x", append "es" to the word.
4. Else append "s" to the word.

Input Specification

The first line of the input consists of two integers **L** and **N** (*0 <= L <= 20*, *1 <= N <= 100*). The following **L** lines contain the description of the irregular words and their plural form. Each line consists of two words separated by a space character, where the first word is the singular, the second word the plural form of some irregular word. After the list of irregular words, the following **N** lines contain one word each, which you have to make plural. You may assume that each word consists of at most 20 lowercase letters from the english alphabet ('a' to 'z').

Output Specification

Print **N** lines of output, where the $\mathbf{i}^{\mathbf{th}}$ line is the plural form of the $\mathbf{i}^{\mathbf{th}}$ input word.

Sample Input

```
3 7
rice rice
spaghetti spaghetti
octopus octopi
rice
lobster
spaghetti
strawberry
octopus
peach
turkey
```

Sample Output

```
rice
lobsters
spaghetti
strawberries
octopi
peaches
turkeys
```

Added by:	Adrian Kuegel
Date:	2007-07-06
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	University of Ulm Local Contest 2007

SPOJ Problem Set (tutorial)

1756. Find The Determinant

Problem code: DETER

In this problem you have to calculate the determinant of an N x N matrix whose entries are given by **m[i][j] = gcd(i,j)**, 1 <= i,j <= N.
Here gcd(i,j) denotes the greatest common divisor of i and j.
As the determinant D can grow very large, you have to print D%1000003.

Input

First line of input consists of a single integer containing the number of test cases T (equal to around 500000), each of the following T lines contain an integer N the size of the matrix. N lies between 1 and 2000000 (both inclusive).

Output

One line corresponding to each test case containing the determinant modulo 1000003 for the corresponding test case.

Example

```
Input:
3
1
3
5

Output:
1
2
16
```

Added by: Ajay Somani
Date: 2007-09-01
Time limit: 6s
Source limit: 2048B
Languages: All
Resource: "The Art Of Computer Programming"

SPOJ Problem Set (tutorial)

1872. Making Book

Problem code: MKBOOK

A printer - who still uses moveable type - is preparing to print a set of pages for a book. These pages are to be numbered, as usual. The printer needs to know how many instances of each decimal digit will be required to set up the page numbers in the section of the book to be printed.

For example, if pages 10, 11, 12, 13, 14 and 15 are to be printed, computing the number of digits is relatively simple: just look at the page numbers that will appear, and count the number of times each digit appears. The digit 0 appears only once, the digit 1 appears 7 times, the digits 2, 3, 4 and 5 each appear once, and 6, 7, 8 and 9 don't appear at all.

Your task in this problem is to provide the printer with the appropriate counts of the digits. You will be given the numbers of the two pages that identify the section of the book to be printed. You may safely assume that all pages in that section are to be numbered, that no leading zeroes will be printed, that page numbers are positive, and that no page will have more than three digits in its page number.

Input

There will be multiple cases to consider. The input for each case has two integers, *A* and *B*, each of which is guaranteed to be positive. These identify the pages to be printed. That is, each integer *P* between *A* and *B*, including *A* and *B*, is to be printed. A single zero will follow the input for the last case.

Output

For each input case, display the case number (1, 2, ...) and the number of occurrences of each decimal digit 0 through 9 in the specified range of page numbers. Display your results in the format shown in the examples below.

Example

```
Input:
10 15
912 912
900 999
0

Output:
Case 1: 0:1 1:7 2:1 3:1 4:1 5:1 6:0 7:0 8:0 9:0
Case 2: 0:0 1:1 2:1 3:0 4:0 5:0 6:0 7:0 8:0 9:1
Case 3: 0:20 1:20 2:20 3:20 4:20 5:20 6:20 7:20 8:20 9:120
```

Added by:	Camilo Andrés Varela León
Date:	2007-10-07
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	North Central North America Regional Programming Contest - 2003

SPOJ Problem Set (tutorial)

2018. Clique Separation

Problem code: CLIQSEP

The Clique Problem

Problem

Let **G** be the set of di-graphs with **n** nodes, **m** edges and maximum clique (complete subgraph) size of **k** nodes, determine whether it is possible to divide every element of **G** into two disjoint sets of nodes, such that the largest size of a clique contained in one set is equal to the largest size of a clique contained in the other set.

The Input

Each line of input has **n <= 1000 , m <= 1000000 , k <= n** , listed in that order.

The Output

For each line of input, output "yes" if it is possible, "no" if it is not possible.

Sample Input

```
10 99 8
9 80 3
```

Sample Output

```
yes
no
```

Problemsetter --- Chen, Xiaohong

Added by: Chen Xiaohong
Date: 2007-11-06
Time limit: 1s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

2020. Painting Points

Problem code: PAINTPOI

Painting Points

Problem

Two players play the following game. The first player paints a point on the plane red. The second player paints **k** uncoloured points on the plane green. The first player paints an uncoloured point on the plane red. The second player paints **k** uncoloured points on the plane green. And so on. The first player wins if there are three red points which form an equilateral triangle. The second player wins if it is not possible within a finite number of moves. Assume he plays prefectly to prevent or delay the first player from winning. Given **k**, determine the minimum number of moves it takes for the first player to force a win. If it's not possible for the first player to win, output **-1**.

The Input

Each line of input has an even integer k, **0 < k <= 1000000**.

The Output

For each line of input, output the answer on one line.

Sample Input

```
10
```

Sample Output

```
12
```

Problemsetter --- Wu, Xiaogang

Added by: Chen Xiaohong
Date: 2007-11-06
Time limit: 1s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

2122. Billboard

Problem code: BBOARD

The manager of the International Popcorn-Selling Company has just decided that a number of advertising billboards should be installed throughout the city. The city consists of a number of crossings connected by (bidirectional) streets. Crossings are numbered by integers 1..N.

There should be one billboard at every crossing. However, to cut down expenses, there have been only three types of billboards printed. Nevertheless, the billboards should be arranged in such a way that one never meets the same billboard twice in a row when driving through the city (suppose that it is possible to turn back only at the crossing). How should they be installed?

Input specification

The input file starts with a line containing the number of test cases. Every test case starts with a line containing two (blank separated) integers N($1<= N <=600$), M($1<= M <=10000$) where N is the number of crossings and M is the number of streets. Each of the next M lines contains two integers x, y, indicating a street connecting crossings x and y.

Output specification

The output file contains a sequence of N numbers delimited by whitespace for every test case. The i-th member of this sequence denotes the type of the billboard at the crossing i (assume that the types of the billboards are numbered 1,2,3). If it is not possible to install the billboards in the described manner, the sequence consists of a single number -1.

Note that it is not necessary to write the entire sequence in one line. To prevent the problems with the mailer you may split long lines into several shorter ones.

Example

Input file:
```
2
6 7
1 3
1 4
5 2
2 6
4 2
3 4
6 3
5 8
1 2
1 5
1 3
2 5
2 3
5 3
3 4
```

```
4 5
```

Output file:

```
1 2 2 3 3 1
-1
```

Added by:	Blue Mary
Date:	2007-12-01
Time limit:	3s
Source limit:	50000B
Languages:	All except: C99 strict
Resource:	IPSC 1999

SPOJ Problem Set (tutorial)

2134. Colorful Cubes

Problem code: CCUBE

Bill Games, an excellent programmer, spent Easter with his grandparents. In their old house he came across wooden cubes - a child's toy. When he was a child, he used to build castles and towers and pyramids of these colorful cubes. He started to play with them again. However, the problem which he tries to solve today is much more complicated than building a simple pyramid.

Each face of a cube is colored with one color. Bill wants to build a tower from all cubes he has. This means to stack all the cubes in one column, one on another. Bill does not want to put the cubes in arbitrary order - the bottom face of every cube (except the bottom cube which is lying on the floor) should have the same color as the top face of the cube below it.

Input file specification

The first line of the input file consists of two numbers M(1<=M<=100) and N(1<=N<=500). M is the number of colors used (colors are numbered 1...M) and N is the number of cubes (cubes are numbered 1...N in the order as they appear on the input).

Next N lines represent cubes 1,2,...,N in this order. A cube is described by six numbers giving colors of its faces in the following order: front, back, right, left, bottom, and top face.

Output file specification

Given the cubes described in the input file determine how to arrange them into a tower. Every cube has to be used exactly once. You need to find only one solution. Assume that the solution exists.

The output file consists of N lines. The i-th line contains the description of the cube on the i-th position in the tower, counting from bottom. The description of a cube consists of seven numbers. The first number is the number of the cube (the order of the cube in the input file) and the following six numbers represent colors of the faces in the following order: front, back, right, left, bottom, and top face. Notice that cubes can be rotated.

Example

Input file #1:
```
6 2
1 2 3 4 5 6
2 1 3 4 5 6
```

Output file #1:
```
1 6 5 3 4 1 2
2 6 5 3 4 2 1
```

Input file #2:
```
3 3
1 2 2 2 1 2
3 3 3 3 3 3
```

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

Output file #2:

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

Added by:	Blue Mary
Date:	2007-12-01
Time limit:	30s
Source limit:	10000B
Languages:	All except: C99 strict
Resource:	IPSC 2000

SPOJ Problem Set (tutorial)

2155. Jamcode 2006 (Easy)

Problem code: JCEASY

There is one unnamed popular programming contest for people from all around the world. (Its name matches "SearchEngine Program Marmalade".) The contest starts with a coding phase where the contestants write code. After the coding phase there is a challenge phase. In this phase one can gain points when she finds a bug in someone else's code.

We were all lame and performed very badly. In fact, none of our programs worked. Thus we decided to hold a new contest: the Jam Code. Here, the task is to write a program that will never work correctly.

This contest will have an anti-challenge phase, where your goal is to find at least one test case such that a given program actually works ; in other words, it computes the correct answer.

Problem specification

You will be given a programming task and someone's source code. Find a valid input such that the program computes the correct answer.

Easy Task specification

The input file contains an integer **M** (0< **M** and **M** < 200) followed by **M** integers a[1], ..., a[m] in the range 1, 2, ..., 334. Output one line with the string $s_{a[1]} s_{a[2]} ... s_{a[m]}$. Here is the list of strings s_1, ..., s_{334}.

Example

```
Input
3
1
2
3

Output
020202020202020212021202121212021202121202021212
```

The file jceasy.cpp contains the program you are supposed to anti-challenge.

You are to submit a file which contains a valid input.

Added by:	Blue Mary
Date:	2007-12-01
Time limit:	1s
Source limit:	50000B
Languages:	TEXT
Resource:	IPSC 2006

SPOJ Problem Set (tutorial)

2156. Jamcode 2006 (Hard)

Problem code: JCHARD

There is one unnamed popular programming contest for people from all around the world. (Its name matches "SearchEngine Program Marmalade".) The contest starts with a coding phase where the contestants write code. After the coding phase there is a challenge phase. In this phase one can gain points when she finds a bug in someone else's code.

We were all lame and performed very badly. In fact, none of our programs worked. Thus we decided to hold a new contest: the Jam Code. Here, the task is to write a program that will never work correctly.

This contest will have an anti-challenge phase, where your goal is to find at least one test case such that a given program actually works ; in other words, it computes the correct answer.

Problem specification

You will be given a programming task and someone's source code. Find a valid input such that the program computes the correct answer.

Hard Task specification

You are organizing a big party for a lot of people. You want to invite **2N** men and **2N** women. At the beginning of the party, there will be a dance. Before the party, each man sent you a list of women he is willing to dance with. You have to maximize the number of pairs that can dance at the same time.

The first line of the input file contains an integer **N** (0< **N** and **N** < 100). On each of the next **2N** lines there are **2N** numbers. If the **i**-th number on the **j**-th line is 0, then the **j**-th man doesn't want to dance with the **i**-th woman. If the number is 1, the man is willing to dance with the woman.

Output one number on one line with the maximum number of pairs which can dance at the same time.

Example

Input
```
1
1 1
1 1
```

Output
```
2
```

The file jchard.cpp contains the program you are supposed to anti-challenge.

You are to submit a file which contains a valid input.

Added by:	Blue Mary
Date:	2007-12-01
Time limit:	1s
Source limit:	50000B
Languages:	TEXT
Resource:	IPSC 2006

SPOJ Problem Set (tutorial)

2261. Program Analyser (tutorial)

Problem code: ANALYS_T

Input

A Program which has the following format:

```
<Program>::=<sentence><line break>{<sentence><line break>}
<setence>::=<level><space><body>
<body>::=<addition> | <output> | <goto> | <condition> | <end>
<addition>::=<variable>+<integer>
<output>::=<variable>?
<goto>::=GO<space><level>
<condition>::=IF<space><variable>=<integer><space><goto>
<end>::=END
<variable>::=<character>
<level>::=<integer>
<integer>::=<digit>{<digit>}
<character>::=A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z
<digit>::= 0|1|2|3|4|5|6|7|8|9
<line break>::=(ASCII 10)
<space>::=(ASCII 32)
```

The program runs following the following rules:

- Program starts from the sentence whose level is minimum, and executed by the level from low to high except that the sentence is<goto>or<condition>.
- All variables are initialized to 0.
- <Addition>means<variable>+=<integer>in C++.
- <output>means write the value of<variable>to the output file(we aren't interesting about the real output file.)
- <condition>means if and only if the value of the <variable> equals to <integer>, <goto> will be executed, otherwise the next sentence executed is as usual.
- After<goto>, the next sentence executed is the sentence with level which equals to the level in<goto>.
- Program terminates by itself when <end> is executed.
- This program can deal with all the signed 32-bit integers.
- The number of sentences in the program is not more than 100.
- The length of each line in the input file is not more than 20.
- The input is correct.
- The sentence with the maximum level is always <end>.
- The levels is not more than 3000.

Input terminate by EOF.

Output

Output the number of sentences executed.If the program can not terminate by itself,output -1.

Example

Input:
```
10 A+1
20 IF A=5 GO 60
60 END
30 A+2
40 A?
50 GO 20
```

Output:
```
11
```

Hint:
```
10->20->30->40->50->20->30->40->50->20->60
```

Added by:	Blue Mary
Date:	2008-01-02
Time limit:	60s
Source limit:	50000B
Languages:	All except: C99 strict
Resource:	a copy of ANALYSER problem with 60s time limit

SPOJ Problem Set (tutorial)

2278. Merrily, We Roll Along!

Problem code: WHEEL

One method used to measure the length of a path is to roll a wheel (similar to a bicycle wheel) along the path. If we know the radius of the wheel and the number of revolutions it makes as it travels along the path, the length of the path can be computed.

This method works well if the path is smooth. But when there are curbs or other abrupt elevation changes in the path, the path distance may not be accurately determined, because the wheel may rotate around a point (like the edge of a curb), or the wheel may roll along a vertical surface. In this problem you are to determine the distance moved by the center of such a wheel as it travels along a path that includes only horizontal and vertical surfaces.

To measure a path, the wheel is placed with its center directly above the origin of the path. The wheel is then moved forward over the path as far as possible, always remaining in contact with the surface, ending with its center directly above the end of the path.

Consider the path shown in the illustration on the left below, and assume the wheel has a radius of 2. The path begins and ends with horizontal segments of length 2 at the same elevation. Between these there is a horizontal segment of length 2.828427 at 2 units below the elevation of the other two horizontal segments. To measure this path, the wheel is placed at position 1. It then moves horizontally to position 2, rotates 45 degrees to position 3, rotates another 45 degrees to position 4, and finally rolls horizontally to position 5. The center of the wheel moved a distance of 7.1416, not 6.8284.

In the illustration on the right below, the path begins and ends with horizontal segments of length 3, separated by a 7-unit wide region placed 7 units below the surface. If the wheel has a radius of 1, then it will move 26.142 units before reaching the end of the path.

[IMAGE]

Input

For this problem there are multiple input cases. Each case begins with a positive real number specifying the radius of the wheel and an integer n, which is at least 1 but not greater than 50. There then follow n pairs of real numbers. The first number in each pair gives the horizontal distance along the path to the next vertical surface. The second number in each pair gives the signed change in the elevation of the path at the vertical surface, with positive numbers representing an increase in elevation. The vertical surfaces are always perpendicular to the horizontal surfaces. The elevation change in the nth pair will always be 0.

The input is terminated by a pair of zeroes.

Output

For each case, display the case number and the distance moved by the center of the wheel with 3 digits to the right of the decimal point.

Place a blank line after the output of each test case.

Example

Input:
```
2.0 3
2.0 -2.0
2.828427 2.0
2.0 0.0
1.0 3
3.0 -7.0
7.0 7.0
3.0 0.0
1.0 3
1.0 -4.0
2.0 4.0
1.0 0.0
0 0
```

Output:
```
Case 1: Distance = 7.142

Case 2: Distance = 26.142

Case 3: Distance = 5.142
```

Added by: Blue Mary
Date: 2008-01-03
Time limit: 1s
Source limit:50000B
Languages: All except: C99 strict
Resource: ACM/ICPC World Final 2002 (unofficial testdata)

SPOJ Problem Set (tutorial)

2533. Point Nesting

Problem code: POINTS

A point in 3D A(ax,ay,az) is said to nest another point B(bx,by,bz), iff bx <= ax AND by <= ay AND bz <= az. Given a set of 3D points, find a nesting sequence using maximal number of points. A sequence P0, P1, P2, ... is said to be a valid nesting sequence iff, P1 nests P0, P2 nests P1 and so on. Please note there could be duplicate points, and each input point must be used atmost once while creating the sequence.

Input

First line contains the number of testcases T.
Each testcase starts with n - The number of points. (0 < n <= 100,000)
The next n lines give the input points.

Output

For each testcase print one integer saying the length of the longest nesting sequence.

Example

```
Input:
2
4
930887 692778 636916
747794 238336 885387
760493 516650 641422
202363 490028 368691
10
897764 513927 180541
383427 89173 455737
5212 595369 702568
956430 465783 21531
722863 665124 174068
703136 513930 979803
634023 723059 133070
898168 961394 18457
175012 478043 176230
377374 484422 544920

Output:
2
3
```

Added by:	Prasanna
Date:	2008-03-12
Time limit:	3s
Source limit:	50000B
Languages:	All
Resource:	CMI Local Contest

SPOJ Problem Set (tutorial)

2625. Problems Collection (Volume 3)

Problem code: PCV3

Problem 1 Using a combination of black square tiles and oblong tiles chosen from: red tiles of two units length, green tiles of three units length, and blue tiles of four units length, it is possible to tile a row of five units length in exactly fifteen different ways:

[IMAGE][IMAGE][IMAGE][IMAGE][IMAGE]	[IMAGE][IMAGE][IMAGE][IMAGE]	[IMAGE][IMAGE][IMAGE][IMAGE]	[IMAGE][IMAGE][IMAGE][IMAGE]
[IMAGE][IMAGE][IMAGE][IMAGE]	[IMAGE][IMAGE][IMAGE]	[IMAGE][IMAGE][IMAGE]	[IMAGE][IMAGE][IMAGE]
[IMAGE][IMAGE][IMAGE]	[IMAGE][IMAGE][IMAGE]	[IMAGE][IMAGE][IMAGE]	[IMAGE][IMAGE]
[IMAGE][IMAGE]	[IMAGE][IMAGE]	[IMAGE][IMAGE]	

In how many ways can a row measuring fifty units in length be tiled?

Problem 2 A hexagonal tile with number 1 is surrounded by a ring of six hexagonal tiles, starting at "12 o'clock" and numbering the tiles 2 to 7 in an anti-clockwise direction. New rings are added in the same fashion, with the next rings being numbered 8 to 19, 20 to 37, 38 to 61, and so on. The diagram below shows the first three rings:

[IMAGE]

By finding the difference between tile n and each of its six neighbours we shall define PD(n) to be the number of those differences which are prime. For example, working clockwise around tile 8 the differences are 12, 29, 11, 6, 1, and 13. So PD(8) = 3. In the same way, the differences around tile 17 are 1, 17, 16, 1, 11, and 10, hence PD(17) = 2. It can be shown that the maximum value of PD(n) is 3. If all of the tiles for which PD(n) = 3 are listed in ascending order to form a sequence, the 10th tile would be 271. Find the 2000th tile in this sequence.

Problem 3 Let S(A) represent the sum of elements in set A of size n. We shall call it a special sum set if for any two non-empty disjoint subsets, B and C, the following properties are true:
1) S(B) != S(C); that is, sums of subsets cannot be equal.
2) If B contains more elements than C then S(B) > S(C).
For this problem we shall assume that a given set contains n strictly increasing elements and it already satisfies the second rule.
Surprisingly, out of the 25 possible subset pairs that can be obtained from a set for which n = 4, only 1 of these pairs needs to be tested for equality (first rule). Similarly, when n = 7, only 70 out of the 966 subset pairs need to be tested. For n = 12, how many of the 261625 subset pairs that can be obtained need to be tested for equality?

Problem 4 Find the smallest integer N > 15, for which N^3 can be written using prime digits only {i.e., 2, 3, 5, 7}.

Problem 5 Let's call an integer a "titanic number" if we need 1000 or more digits to write it in decimal format. In this task you must find the minimal titanic number, which can be presented in p^q form, where p and q are prime numbers. You must output the answer in the following form: X-q, where X - the last 10 digits of the titanic number and q - the power of the exponent. For example: 8765839202-97

Problem 6 Find the smallest positive integer for which every number in the series (N-k)/k is a prime number for every k=1,...n, for n = 11. For n=4 the answer would be N=12, let's check: (12-1)/1 = 11, (12-2)/2 = 5, (12-3)/3 = 3, (12-4)/4 = 2.

Problem 7 You are playing the following game. You can ask the host of the game to tell you a number. Each number is an independent random uniformly distributed real number between 0 and 1. After the host tells you the number you can ask for more or just stop. When you stop, your score is equal to the sum of all numbers which the host has given to you. Let $0 < x < 1$ and suppose that you're trying to get a score in the interval from x to 1. What is the probability of winning, assuming that you are using the best possible strategy? Find the value of probability of winning for x=0.334568 and output it after rounding in the form of *.****** - where each * denotes a digit.

Problem 8 Find the number of integers $1 < n < 10^7$, for which n and n + 1 have the same number of positive divisors. For example, 14 has the positive divisors 1, 2, 7, 14 while 15 has 1, 3, 5, 15.

Problem 9 Decode the message in the picture:

[IMAGE]

Output it with lowercase letters without spaces.

Problem 10 Suppose that you find a small program which is protected by an "activation key". The value of the key depends on the name you input. The protection for this program is performed using the code in the C programming language, presented below. The program asks you for your name and password and outputs "Accept" or "Failure".

```
#include <stdio.h>

unsigned int code (unsigned int arg, int p, int n)
{
        unsigned int r = 1;
        for(; p >= 1; p--)
                r = (r*arg)%n;
        return r;
}

void main ()
{
        unsigned int e = 35467, n = 54031, pwd;
        char name[256];
        unsigned int hash, x;

        printf("Name: ");
        scanf("%s", name);

        printf("Password: ");
        scanf("%d", &pwd);

        hash = 0;
        for (x = 0; ; x++){
                if (name[x] == 0)
                        break;
                hash += name[x];
        }

        if (code(pwd, e, n) == hash)
                printf("Accept!\n");
        else
                printf("Failure\n");
}
```

Your goal is to find the right passwords for each name presented in file: nicks.zip (~330 Kb). The answer for this problem will be the sum of all passwords obtained for each name from file.

Input

There is no input for this problem.

Output

Output answer as a set of lines. In each line first output the number of the problem and then the answer for this problem. If any of the answers are incorrect, you'll receive Wrong Answer.

Score

For each solved problem you'll recieve exactly one point (10 points maximum, if all problems are solved correctly).

Example

Output:

```
1 6174046
2 AnsweR
5 806257
8 51146700
```

It's just an example of what the output should look like. If all 4 answers are correct (for problems 1, 2, 5 and 8), you'll receive 4 points.

Added by: Roman Sol
Date: 2007-07-23
Time limit: 1s
Source limit:10000B
Languages: TEXT
Resource: ZCon 2008

SPOJ Problem Set (tutorial)

2626. RegExp Master

Problem code: REX

You are given a set of 10 tasks. For each of these tasks you must write the correct Regular Expression of minimal size in C format (current SPOJ version). Each regular expression will be tested against a special test set, which contains right and wrong strings. The expression which is composed by you must correctly work on all test cases.

Task 1 You are to write regular expression which determines if a given string is equal to "abcdefghijklmnopqrstuv18340" or not.

<u>Example of correct strings:</u>
abcdefghijklmnopqrstuv18340

<u>Example of wrong strings:</u>
abcdefghijklmnoasdfasdpqrstuv18340

Task 2 You are to write a regular expression which determines whether a given string is a GUID, with or without brackets. Here GUID is a string, consisting of 8, 4, 4, 4, 12 hex digits separated by '-'.

<u>Examples of correct strings:</u>
{e02fa0e4-01ad-090A-c130-0d05a0008ba0}
e02fd0e4-00fd-090A-ca30-0d00a0038ba0

<u>Examples of wrong strings:</u>
02fa0e4-01ad-090A-c130-0d05a0008ba0}
e02fd0e400fd090Aca300d00a0038ba0

Task 3 You are to write a regular expression which determines whether the given string is a valid MAC-address.

<u>Examples of correct strings:</u>
01:32:54:67:89:AB
aE:dC:cA:56:76:54

<u>Examples of wrong strings:</u>
01:33:47:65:89:ab:cd
01:23:45:67:89:Az

Task 4 You are to write a regular expression which determines whether a given string is uppercase and sorted in non-descending order.

<u>Examples of correct strings:</u>
AABCD
ABCDZ

<u>Examples of wrong strings:</u>
aABCD
ZABCD

Task 5 You are to write a regular expression which determines whether a given string is the hex identification of a color in HTML. Here #FFFFFF stands for white, #000000 for black, #FF0000 for red, etc.

<u>Examples of correct strings:</u>
#FFFFFF
#FF3421
#00ff00

<u>Examples of incorrect strings:</u>
232323
f#fddee
#fd2

Task 6 You are to write a regular expression which determines whether the given string is a date in

dd/mm/yyyy format. The date is in the range from the year 1600 to the year 9999.
<u>Examples of correct strings:</u>
29/02/2000
30/04/2003
01/01/2003
<u>Examples of wrong strings:</u>
29/02/2001
30-04-2003
1/1/1899
Task 7 You are to write a regular expression which determines whether the given string is a valid e-mail address with respect to RFC number 2822
<u>Examples of correct strings:</u>
mail@mail.ru
valid@megapochta.com
aa@aa.info
<u>Examples of wrong strings:</u>
bug@@@com.ru
@val.ru
Just Text2
val@val
val@val.a.a.a.a
12323123@111[]][]
Task 8 You are to write a regular expression which determines whether the given string is an IP address, in decimal format
<u>Examples of correct strings:</u>
127.0.0.1
255.255.255.0
192.168.0.1
<u>Examples of wrong strings:</u>
1300.6.7.8
abc.def.gha.bcd
254.hzf.bar.10
Task 9 You are to check whether a given password is strong. A password is said to be strong if it consists of 8 and more symbols, where a symbol is one from the set: English letter, digit or underline. Additionally, a strong password must contain at least one uppercase letter, at least one lowercase letter and at least one digit.
<u>Examples of correct strings:</u>
C00l_Pass
SupperPas1
<u>Examples of wrong strings:</u>
Cool_pass
C00l
Task 10 You are to write a regular expression which determines whether a given string is a six-digit positive integer, printed in decimal format without leading zeros.
<u>Examples of correct strings:</u>
123456
234567
<u>Examples of wrong strings:</u>
1234567
12345
For testing we use the following C-function:

```
int match(const char *string, char *pattern)
{
    int status;
    regex_t re;
```

```
    if (regcomp(&re, pattern, REG_EXTENDED|REG_NOSUB) != 0) {
        return(0);
    }
    status = regexec(&re, string, (size_t) 0, NULL, 0);
    regfree(&re);
    if (status != 0) {
        return (0);
    }
    return (1);
}
```

Input

There is no input data for this problem

Output

Output your answer as a set of 10 lines. The first line is for the first task, the second line for the second task, etc. All other lines will be ignored. If you don't want to solve some task, then in the corresponding line output "---". Otherwise, output the regular expression for this task. If any of your regular expressions are invalid you'll get Wrong Answer status.

Score

For each solved task you'll get exactly 1 point plus a bonus points equal to 1/(regular expression size).

Example

Output:

```
---
^[1-9]{1}[0-9]{3} ?[A-Z]{2}$
---
---
---
---
---
---
---
---
```

It's just an example of what output data should look like. If the answer for second task were right, then you would get 1 + 1/28 = 1.035714 points.

Added by: Roman Sol
Date: 2007-07-31
Time limit: 10s
Source limit: 10000B
Languages: TEXT
Resource: ZCon 2008

SPOJ Problem Set (main)

2627. The Longest Chain

Problem code: BCH

Output the longest chain of integers which has the following properties:

1. All integers are positive and have 4 digits in their decimal representation (i.e. all numbers are in the range [1000, 9999])
2. All numbers in the chain are different
3. The decimal representations of each number differs from the next one at only position (digit)
4. All integers are prime

The winner is the participant who obtains the longest chain.

Input

There is no input data in this problem.

Output

In the first line output the length of your chain *N*. In the next *N* lines output each number of your chain.

Score

The number of points you'll get for the given problem is calculated using following formula: *score* = *1000/(1062 - length)*, where *length* - length of your chain.

Example

```
Output:
3
9857
9887
9883

Score:
In this case score = 1000/(1062-3) = 0.944287,
```

Problem author: Filimonenkov D.O.

Added by:	Roman Sol
Date:	2007-09-03
Time limit:	1s-30s
Source limit:	50000B
Languages:	All
Resource:	ZCon 2008

SPOJ Problem Set (tutorial)

2630. Autoarchive

Problem code: ARJ

Write a program of minimal possible size which outputs the given file: ZARJ.txt (101350 bytes)
Note: it's much better to submit source file, not text, when submit your solution.

Input

There is no input data for this problem

Output

Output your answer in the same format as the given file. If the output is different from the given file then you get status Wrong Answer.

Score

The total number of points obtained for your solution will be equal to its source code size in bytes.

The solution to this problem isn't allowed in some programming languages because these languages support compression functions.

Added by:	Roman Sol
Date:	2008-02-27
Time limit:	20s
Source limit:	100000B
Languages:	C C99 strict C++ PAS gpc PAS fpc JAVA C# ASM
Resource:	Inspired by MAGIC2 (ZCon 2008)

SPOJ Problem Set (main)

2632. Max Power

Problem code: MXP

You are given two sequences of positive integers $a_1, a_2, ..., a_n$ and $b_1, b_2, ..., b_n$ of length *n* each. You are to write a program which finds *k* such that a_k to the power of b_k is maximal.

Input

The first line of input contains a positive integer *n*, not greater than 10000. In the second line you are given a set of positive integers a_i separated by spaces, and in the third line - integers b_i. All numbers in both sequences are not greater than 10000. It is guaranteed that all power values are different.

Output

The output must contain one number - the answer to the problem.

Score

The score to this problem is equal to (1000 - t), where t is the time used by your solution, in milliseconds. If your solution works for more than 1 second then you get 0 points.

Example

```
Input:
5
1 2 2 3 3
100 1 3 2 1

Output:
4
```

Added by: Roman Sol
Date: 2008-03-13
Time limit: 10s-25s
Source limit:50000B
Languages: All
Resource: ZCon 2008

SPOJ Problem Set (tutorial)

2854. El Dorado

Problem code: ELDORADO

Bruce Force has gone to Las Vegas, the El Dorado for gamblers. He is interested especially in one betting game, where a machine forms a sequence of n numbers by drawing random numbers. Each player should estimate beforehand, how many increasing subsequences of length k will exist in the sequence of numbers.

A subsequence of a sequence $a_1, ..., a_n$ is defined as $a_{i_1}, ..., a_{i_l}$, where $1 <= i_1 < i_2 < ... < i_l <= n$. The subsequence is increasing, if $a_{i_{j-1}} < a_{i_j}$ for all $1 < j <= l$.

Bruce doesn't trust the Casino to count the number of increasing subsequences of length k correctly. He has asked you if you can solve this problem for him.

Input

The input contains several test cases. The first line of each test case contains two numbers n and k ($1 <= k <= n <= 100$), where n is the length of the sequence drawn by the machine, and k is the desired length of the increasing subsequences. The following line contains n pairwise distinct integers a_i ($-10000 <= a_i <= 10000$), where a_i is the i^{th} number in the sequence drawn by the machine.

The last test case is followed by a line containing two zeros.

Output

For each test case, print one line with the number of increasing subsequences of length k that the input sequence contains. You may assume that the inputs are chosen in such a way that this number fits into a 64 bit signed integer (in C/C++, you may use the data type "long long", in Java the data type "long").

Example

```
Input:
10 5
1 2 3 4 5 6 7 8 9 10
3 2
3 2 1
0 0
Output:
252
0
```

Added by:	Adrian Kuegel
Date:	2008-07-12
Time limit:	5s
Source limit:	50000B
Languages:	All
Resource:	University of Ulm Local Contest 2008

SPOJ Problem Set (tutorial)

2907. Super Factor Sum

Problem code: FACTSUM

```
Given a positive integer K > 2, with prime
factorization:

K = p1^a1 * p2^a2 ... * pn^an

Compute the following:

S = a1*p1 + a2*p2 ... + an*pn.
```

Input

A list of <100 integers, one on each line, all less than 2*10^19.

Output

For each integer compute the super factor sum and output it on a single line.

Example

```
Input:
6
7
Output:
5
7
```

Added by:	Chen Xiaohong
Date:	2008-08-05
Time limit:	30s
Source limit:	50000B
Languages:	All
Resource:	original

SPOJ Problem Set (tutorial)

2912. Super Primes

Problem code: SPRIME

In mathematics, a prime number (or a prime) is a natural number which has exactly two distinct natural number divisors: 1 and itself.

Super-prime numbers are the elements of the subsequence of prime-numbers that occupy prime-numbered positions within the sequence of all prime numbers. That is, if p(i) denotes the ith prime number, the numbers in this sequence are those of the form p(p(i)) or Primes with a prime index in the sequence of prime numbers (the 2nd, 3rd, 5th, ... prime).

Your task is to generate all super primes ≤ 10^7.

Input:

There is NO input for this problem.

Output:

Print all super-primes ≤ 10^7 in ascending order,one per line.

First few lines of Output

```
3
5
11
17
31
41
59
67
83
109
...
```

Added by: u.swarnaprakash
Date: 2008-08-05
Time limit: 8s
Source limit:10000B
Languages: All
Resource: Myself

SPOJ Problem Set (tutorial)

3032. Adding two numbers

Problem code: ADUN

Your task is to read two numbers a and b ($0 < a, b < 2100000000$) and to output their sum.

Input

Input contains two lines, on the first line the number a and on the second line the number b.

Output

Output the sum of the two numbers.

Example

```
Input:
20
30

Output:
50
```

Added by: Pripoae Toni
Date: 2008-09-14
Time limit: 1s
Source limit: 1024B
Languages: All
Resource: Original

SPOJ Problem Set (tutorial)

3081. Look and Say

Problem code: LOOKSAY

The look and say sequence is defined as follows. Start with any string of digits as the first element in the sequence. Each subsequent element is defined from the previous one by "verbally" describing the previous element. For example, the string 122344111 can be described as "one 1, two 2's, one 3, two 4's, three 1's". Therefore, the element that comes after 122344111 in the sequence is 1122132431. Similarly, the string 101 comes after 1111111111. Notice that it is generally not possible to uniquely identify the previous element of a particular element. For example, a string of 112213243 1's also yields 1122132431 as the next element.

Input

The input consists of a number of cases. The first line gives the number of cases to follow. Each case consists of a line of up to 1000 digits.

Output

For each test case, print the string that follows the given string.

Example

```
Input:
3
122344111
1111111111
12345

Output:
1122132431
101
1112131415
```

Added by: Nikola P Borisov
Date: 2008-10-01
Time limit: 4s-30s
Source limit:50000B
Languages: All
Resource: ICPC North America Rocky Mountain Regional Contest 2007

SPOJ Problem Set (tutorial)

3131. Time to Graduate

Problem code: CURICULA

Consider the following example. A student is required to take 4 courses, mt42, cs123, cs456, and cs789. mt42 is only offered in the fall semester and has no prerequisites. Similarly, cs123 is only offered in the spring semester and has no prerequisites. cs456 is only offered in the spring semester and has both cs123 and mt42 as prerequisites. Finally, cs789 is offered in both fall and spring and has cs456 as its only prerequisite. The shortest time to graduate is 5 semesters, by taking mt42 in the fall, cs123 in the next spring, cs456 the following spring (since it is not offered in the fall) and finally cs789 the following fall.

For this problem, there are only two semesters, fall and spring. Always start counting semesters from the fall.

In addition to the fall/spring scheduling issues, there is one slight complication. In order to keep the dormitories full, each university limits the number of courses that can be taken in any semester. This limit appears as part of the input data. The third example below illustrates this issue.

Input

There are one to twenty-five data sets, followed by a final line containing only the integers '`-1 -1`'. A data set starts with a line containing two positive integers n , 1 <= n <= 12 , which is the number of courses in this data set and m , 2 <= m <= 6 , which is the maximum number of courses that can be taken in any single semester. The next line contains the n course identifiers. Each is a 1-5 character string from the set `{a-z, 0-9}`. Following the course identifiers is the individual course information. This consists of n lines, one line for each course, containing the course identifier, semester offered ('`F`'=Fall, '`S`'=Spring, '`B`'=Both semesters), the number of prerequisite courses, p , 0 <= p <= 5 , and finally p prerequisite course identifiers. The first example data set below corresponds to the problem described above.

Output

The output contains one line for each data set, formatted as shown in the sample output.

Example

Input:

```
4 6
cs123 mt42 cs456 cs789
mt42 F 0
cs123 S 0
cs456 S 2 cs123 mt42
cs789 B 1 cs456
3 6
math1 comp2 comp3
comp3 S 1 comp2
math1 S 0
comp2 F 1 math1
```

```
4 3
m10 m20 c33 c44
m10 B 0
m20 B 0
c33 B 0
c44 B 0
-1 -1
```

Output:

```
The minimum number of semesters required to graduate is 5.
The minimum number of semesters required to graduate is 4.
The minimum number of semesters required to graduate is 2.
```

Added by:	Nikola P Borisov
Date:	2008-10-11
Time limit:	5s
Source limit:	50000B
Languages:	All
Resource:	Mid-Central Regional ACM-ICPC Contest 2005

SPOJ Problem Set (tutorial)

3250. The Seven Percent Solution

Problem code: SEVENPER

Uniform Resource Identifiers (or URIs) are strings like *http://icpc.baylor.edu/icpc/*, *mailto:foo@bar.org*, *ftp://127.0.0.1/pub/linux*, or even just *readme.txt* that are used to identify a resource, usually on the Internet or a local computer. Certain characters are reserved within URIs, and if a reserved character is part of an identifier then it must be *percent-encoded* by replacing it with a percent sign followed by two hexadecimal digits representing the ASCII code of the character. A table of seven reserved characters and their encodings is shown below. Your job is to write a program that can percent-encode a string of characters.

Character	**Encoding**
" " (space)	%20
"!" (exclamation point)	%21
"$" (dollar sign)	%24
"%" (percent sign)	%25
"(" (left parenthesis)	%28
")" (right parenthesis)	%29
"*" (asterisk)	%2a

Input

The input consists of one or more strings, each 1-79 characters long and on a line by itself, followed by a line containing only "#" that signals the end of the input. The character "#" is used only as an end-of-input marker and will not appear anywhere else in the input. A string may contain spaces, but not at the beginning or end of the string, and there will never be two or more consecutive spaces.

Output

For each input string, replace every occurrence of a reserved character in the table above by its percent-encoding, exactly as shown, and output the resulting string on a line by itself. Note that the percent-encoding for an asterisk is %2a (with a lowercase "a") rather than %2A (with an uppercase "A").

Example

Input:
```
Happy Joy Joy!
http://icpc.baylor.edu/icpc/
plain_vanilla
(**)
```

```
?
the 7% solution
#
```

Output:

```
Happy%20Joy%20Joy%21
http://icpc.baylor.edu/icpc/
plain_vanilla
%28%2a%2a%29
?
the%207%25%20solution
```

Added by:	Nikola P Borisov
Date:	2008-10-25
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	Mid-Central Regional ACM-ICPC Contest 2007

SPOJ Problem Set (tutorial)

3252. Persistent Bits

Problem code: PERBITS

WhatNext Software creates sequence generators that they hope will produce fairly random sequences of 16-bit unsigned integers in the range 0-65535. In general a sequence is specified by integers A, B, C, and S, where 1 <= A < 32768, 0 <= B < 65536, 2 <= C < 65536, and 0 <= S < C. S is the first element (the *seed*) of the sequence, and each later element is generated from the previous element. If X is an element of the sequence, then the next element is

(A*X + B) % C

where '%' is the remainder or modulus operation. Although every element of the sequence will be a 16-bit unsigned integer less than 65536, the intermediate result A*X + B may be larger, so calculations should be done with a 32-bit *int* rather than a 16-bit *short* to ensure accurate results.

Some values of the parameters produce better sequences than others. The most embarrassing sequences to WhatNext Software are ones that never change one or more bits. A bit that never changes throughout the sequence is *persistent*. Ideally, a sequence will have no persistent bits. Your job is to test a sequence and determine which bits are persistent.

For example, a particularly bad choice is A = 2, B = 5, C = 18, and S = 3. It produces the sequence 3, (2*3+5)%18 = 11, (2*11+5)%18 = 9, (2*9+5)%18 = 5, (2*5+5)%18 = 15, (2*15+5)%18 = 17, then (2*17+5)%18 = 3 again, and we're back at the beginning. So the sequence repeats the the same six values over and over:

Decimal	16-Bit Binary
3	0000000000000011
11	0000000000001011
9	0000000000001001
5	0000000000000101
15	0000000000001111
17	0000000000010001
overall	00000000000????1

The last line of the table indicates which bit positions are always 0, always 1, or take on both values in the sequence. Note that 12 of the 16 bits are persistent. (Good random sequences will have no persistent bits, but the converse is not necessarily true. For example, the sequence defined by A = 1, B = 1, C = 64000, and S = 0 has no persistent bits, but it's also not random: it just counts from 0 to 63999 before repeating.) Note that a sequence does not need to return to the seed: with A = 2, B = 0, C = 16, and S = 2, the sequence goes 2, 4, 8, 0, 0, 0, ...

Input

There are from one to sixteen datasets followed by a line containing only 0. Each dataset is a line containing decimal integer values for A, B, C, and S, separated by single blanks.

Output

There is one line of output for each data set, each containing 16 characters, either '1', '0', or '?' for each of the 16 bits in order, with the most significant bit first, with '1' indicating the corresponding bit is always 1, '0' meaning the corresponding bit is always 0, and '?' indicating the bit takes on values of both 0 and 1 in the sequence.

Example

Input:
```
2 5 18 3
1 1 64000 0
2 0 16 2
256 85 32768 21845
1 4097 32776 248
0
```

Output:
```
00000000000????1
????????????????
000000000000???0
0101010101010101
0???000011111???
```

Added by: Nikola P Borisov
Date: 2008-10-25
Time limit: 1s
Source limit:50000B
Languages: All
Resource: Mid-Central Regional ACM-ICPC Contest 2007

SPOJ Problem Set (tutorial)

3256. Rock Skipping

Problem code: ROCKSKIP

As a member of the International Rock-Skipping League, you travel to a different lake each week and compete in a rock-skipping contest. The goal is to throw a rock so that it skips as many times as possible; the exact rules for determining the winner are given below. To make the competitions interesting, the IRSL often chooses lakes with logs, sandbars, and other obstacles. You are provided with a side-view, water-level "map" of the lake as shown in the top line of the example below. (The numbers 0..29 below the map are just for reference.) A period (".") indicates clear water, where a rock will skip; any other character indicates some kind of obstacle that will stop a rock.

```
...=...**..#...@....:.:.:..=..
______________________________
          11111111112222222222
012345678901234567890123456789
```

You stand at the left end of the lake. You can throw a rock so that it lands at any position in the lake, and then skips at any fixed interval thereafter. So a throw can be defined as a pair (i,d), where $i >= 0$ is the initial landing position and $d > 0$ is the distance between skips. Note that d must be positive. The *count* of a throw is the number of times that it skips on the water. The *length* is the position of its last contact with either the water or an obstacle. To rank two distinct throws, use the following criteria, in order, until a winner is determined: count (highest wins); length (greatest wins); initial position (greatest wins); distance between skips (smallest wins).

For the map shown above, throw (27,2) hits the obstacle at position 27; it has count 0 and length 27. Throw (16,1) skips at positions 16, 17, 18, and 19, then hits the obstacle at position 20; it has count 4 and length 20, so it beats throw (27,2). Throw (2,7) skips at positions 2, 9, 16, and 23, then skips over the lake; it has count 4 and length 23, so it beats throw (16,1). Throw (1,4) skips at positions 1, 5, 9, 13, 17, 21, 25, and 29, then skips over the lake; it has count 8 and distance 29, and is the best possible throw for this lake.

Input

The input consists of one or more lake maps, each 1-40 characters long and on a line by itself, followed by a line containing only "END" that signals the end of the input. Positions within a map are numbered starting with zero. Maps will only contain printable ASCII punctuation characters. A period indicates clear water and any other character indicates an obstacle.

Output

For each map, compute the best possible throw (i,d), then output a line containing i and d separated by one space.

Example

Input:

```
...=...**..#...@....:.:.:..=..
.(+)
/^\.
*++&*
END
```

Output:

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

Added by: Nikola P Borisov
Date: 2008-10-25
Time limit: 1s
Source limit: 50000B
Languages: All
Resource: Mid-Central Regional ACM-ICPC Contest 2007

SPOJ Problem Set (tutorial)

3313. Software Bugs

Problem code: SOFTBUG

Problem text

The biggest problem for all software developers are bugs. You definitely know the situationwhen a user calls to say "I've found a bug in your program". Once you have found and removedthe bug, another one appears immediately. It is a hard and never-ending process.Recently, there appeared a promising open-source initiative called "bug-preprocessor". The bug-preprocessor is a program able to find all bugs in your source code and mark them, so they arerelatively easy to be removed. Your task is to write a program that will remove all marked bugsfrom the preprocessed source code.

Input Specification

The input contains a text representing the preprocessed source code, an unspecified number oflines of text, some of them may be empty. Bugs are represented by a case-sensitive string "BUG".The text is terminated by the end of file. No line in the input will be longer than 100 characters.

Output Specification

Your program must remove all of the bugs from the input and print a text that does not containany BUG strings. Nothing else than bugs may be removed, not even spaces.

Example

Input: print "No bugs here..." void hello() { BUGBUG printfBUG("Hello, world!\n"); } wriBUGBUGtelBUGn("Hello B-U-G");

Output: print "No bugs here..." void hello() { printf("Hello, world!\n"); } writeln("Hello B-U-G");

Added by: Robert Rychcicki
Date: 2008-11-07
Time limit: 10s
Source limit:50000B
Languages: C C99 strict C++ PAS gpc PAS fpc JAVA C# PERL PYTH TEXT
Resource: CEPC 2007

SPOJ Problem Set (tutorial)

3318. Pascals Travels

Problem code: PASCALTR

An $n \times n$ game board is populated with integers, one nonnegative integer per square. The goal is to travel along any legitimate path from the upper left corner to the lower right corner of the board. The integer in any one square dictates how large a step away from that location must be. If the step size would advance travel off the game board, then a step in that particular direction is forbidden. All steps must be either to the right or toward the bottom.

Consider the 4 x 4 board shown in Figure 1, where the solid circle identifies the start position and the dashed circle identifies the target. Figure 2 shows the three paths from the start to the target, with the irrelevant numbers in each removed.

$\textstyle \parbox{.24\textwidth}{ \begin{center} \mbox{} \epsfxsize=1.5in \epsfbox{p3390a.eps} \par \medskip Figure 1 \end{center}}$ $\textstyle \parbox{.75\textwidth}{ \begin{center} \mbox{} \epsfxsize=4.5in \epsfbox{p3390b.eps} \par \medskip Figure 2 \end{center}}$

Figure 1 Figure 2

Input

The input contains data for one to thirty boards, followed by a final line containing only the integer '`-1`'. The data for a board starts with a line containing a single positive integer n , 4 <= n <= 34 , which is the number of rows in this board. This is followed by n rows of data. Each row contains n single digits, 0-9, with no spaces between them..

Output

The output consists of one line for each board, containing a single integer, which is the number of paths from the upper left corner to the lower right corner. There will be fewer than 2^{63} paths for any board.

Warning: Brute force methods examining every path will likely exceed the allotted time limit. 64-bit integer values are available as long values in Java or long long values in C/C++ compilers.

Example

```
Input:
4
2331
1213
1231
3110
4
3332
1213
1232
2120
```

```
5
11111
11111
11111
11111
11111
-1
```

Output:

```
3
0
70
```

Added by: Nikola P Borisov
Date: 2008-11-09
Time limit: 10s
Source limit:50000B
Languages: All
Resource: Mid-Central Regional ACM-ICPC Contest 2005

SPOJ Problem Set (tutorial)

3319. Speed Limit

Problem code: SPEEDLM

Bill and Ted are taking a road trip. But the odometer in their car is broken, so they don't know how many miles they have driven. Fortunately, Bill has a working stopwatch, so they can record their speed and the total time they have driven. Unfortunately, their record keeping strategy is a little odd, so they need help computing the total distance driven. You are to write a program to do this computation.

For example, if their log shows

Speed in miles per hour	Total elapsed time in hours
20	2
30	6
10	7

this means they drove 2 hours at 20 miles per hour, then 6-2=4 hours at 30 miles per hour, then 7-6=1 hour at 10 miles per hour. The distance driven is then (2)(20) + (4)(30) + (1)(10) = 40 + 120 + 10 = 170 miles. Note that the total elapsed time is always since the beginning of the trip, not since the previous entry in their log.

Input

The input consists of one or more data sets. Each set starts with a line containing an integer *n*, 1 <= n <= 10, followed by *n* pairs of values, one pair per line. The first value in a pair, s, is the speed in miles per hour and the second value, t, is the total elapsed time. Both s and t are integers, 1 <= s <= 90 and 1 <= t <= 12. The values for t are always in strictly increasing order. A value of -1 for n signals the end of the input.

Output

For each input set, print the distance driven, followed by a space, followed by the word "miles".

Example

```
Input:
3
20 2
30 6
10 7
2
60 1
30 5
4
15 1
25 2
30 3
```

```
10 5
-1
```

Output:

```
170 miles
180 miles
90 miles
```

Added by:	Nikola P Borisov
Date:	2008-11-09
Time limit:	10s
Source limit:	50000B
Languages:	All
Resource:	Mid-Central Regional ACM-ICPC Contest 2004

SPOJ Problem Set (tutorial)

3320. Longest Monotonically Nondecreasing Sequence

Problem code: LMIS

You are given a set of numbers on the standard input. You need to figure out what is the smallest number of them that you need to remove so that you are left with the Longest Monotonically Nondecreasing Sequence.

Input

On the standard input your are given a set with less than 1000000 integers each less than 30000.

Output

A single integer - the number of numbers you will remove.

Example

```
Input:
3 1 2 0 5 4 10

Output:
3
```

(you need to remove 3, 0, (5 or 4), then you will be left with 1, 2, (5 or 4), 10).

Added by: Nikola P Borisov
Date: 2008-11-09
Time limit: 15s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

3321. The Knapsack Problem

Problem code: KNAPSACK

The famous knapsack problem. You are packing for a vacation on the sea side and you are going to carry only one bag with capacity S (1 <= S <= 2000). You also have N (1<= N <= 2000) items that you might want to take with you to the sea side. Unfortunately you can not fit all of them in the knapsack so you will have to choose. For each item you are given its size and its value. You want to maximize the total value of all the items you are going to bring. What is this maximum total value?

Input

On the first line you are given S and N. N lines follow with two integers on each line describing one of your items. The first number is the size of the item and the next is the value of the item.

Output

You should output a single integer on one like - the total maximum value from the best choice of items for your trip.

Example

```
Input:
4 5
1 8
2 4
3 0
2 5
2 3


Output:
13
```

Added by: Nikola P Borisov
Date: 2008-11-10
Time limit: 1s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

3370. Mergesort

Problem code: MERGSORT

Simple. Sort the numbers on the standard input using the merge sort algorithm. Don't try to cheat by just calling your build in functions... I can see your source.

Input

On the standard input you will receive N (1 <= N <= 100000). Each number will fit in 32-bit integer

Output

Output the same integers in a sorted manner. Smallest to largest.

Example

```
Input:
7 3 2 5 4 3

Output:
2 3 3 4 5 7
```

Added by: Nikola P Borisov
Date: 2008-11-17
Time limit: 1s-2s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

3371. Fast Power

Problem code: FASTPOW

You are asked to compute A^B (A to the power of B) (1 <= A,B <= 100 000 000). You surely understand that this number can be quite astonishing and i don't like big numbers unless they are on my paycheck so I'm just interested in the reminder of A^B mod C (1<= C <= 1000000)

Input

Three integers A B and C.

Output

Single integer - the result of the computation.

Example

```
Input:
2 10 1000

Output:
24
```

Added by: Nikola P Borisov
Date: 2008-11-17
Time limit: 1s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

3421. Garden Hull

Problem code: GARDENHU

We have a garden with trees in it. For each tree is represented by x and y coordinates. Your goal is to find the least amount of net needed to surround all the trees so that all trees are inside the net. In other words you need to find the the length of the convex hull around those points

Input

On the first line a lonely integer N (3 <= N <= 10000) representing the number of trees in the garden. On each of the following N lines you will find two integers - the coordinates of the next tree.

Output

A single integer - the length of the convex hull. Round it up to an integer.

Example

```
Input:
4
0 0
5 0
1 1
0 5


Output:
17
```

Added by:	Nikola P Borisov
Date:	2008-11-24
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	Fall Bulgarian National Contest 2003

SPOJ Problem Set (tutorial)

3422. Calculate the Area

Problem code: CALCAREA

Very simple. You are given polygon and you need to calculate his area. The polygon does not self intersect

Input

A lonely integer N (3 <= N <= 10000) - the number of nodes. On each of the next N lines you will find the coordinates of the next vertex from the polygon.

Output

A single integer - the rounded up area of the polygon.

Example

```
Input:
4
0 0
1 0
1 1
0 1

Output:
1
```

Added by: Nikola P Borisov
Date: 2008-11-24
Time limit: 1s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

3440. Enormous Input and Output Test

Problem code: INOUTEST

Similar to that of the problem Enormous Input Test, the purpose of this problem is to determine whether your method of reading input data and writing output data is fast enough to process extremely large test cases.

Input

The first line of input contains a single integer N ($1 <= N <= 10^6$), denoting the number of lines to follow. Each of these lines contains two space-separated integers a and b (|a|, |b| <= 40,000).

Output

For each pair a and b, write a single line of output containing the value of a × b.

Example

```
Input:
5
-1 -1
1 1
0 999
654 321
39999 -39999

Output:
1
1
0
209934
-1599920001
```

Added by: Neal Wu
Date: 2008-11-30
Time limit: 20s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

3443. Netiquette

Problem code: NETIQUE

Mr. Manners teaches netiquette ("net etiquette", particularly as it applies to email) at the local community college. There are many different aspects to proper netiquette, including courtesy, correct spelling, and correct grammar. Through experience Mr. Manners has found that his college's email system does a good job of catching most spelling and grammatical errors, and he's also found that most of his students are courteous. So there are four violations of netiquette that Mr. Manners pays careful attention to, and he's devised a quick way to test for them. A message is *suspicious* if it contains any of the following:

1. two adjacent uppercase letters,
   (because you might be SHOUTING)
2. a digit adjacent to a letter,
   (because you might be l33t, d00d)
3. an isolated character other than *a*, *A*, or *I*,
   (because u r probably abbreviating words; the spell checker doesn't catch this for some reason)
4. two adjacent punctuation marks, unless one of them is a double quote (the character ").
   (because you might be using an emoticon :-)

For this problem, all characters in an email message are printable ASCII characters with codes in the range 32..126 (inclusive). A *punctuation mark* is any character other than a letter, digit, or space. Two characters are *adjacent* if they are right next to each other, with no characters in between. An *isolated character* is one whose only adjacent characters (if any) are spaces. Your job is to write a program that can tell if a one-line email message is suspicious.

Input

The input consists of one or more email messages, followed by a line containing only # that signals the end of the input. Each message is on a line by itself, does not begin or end with a space, and does not contain consecutive spaces. End-of-line characters occur at the end of every line (of course), but they are not considered to be part of the message. A message will contain 1..80 characters.

Output

For each message, output *suspicious* if it meets one or more of the four criteria defined above, and output *OK* otherwise. In the examples below, the second email meets all four criteria, and the fourth and sixth emails meet one criterion each.

Example

```
Input:
"This is a safe message," said 50 wise men.
DON'T b l8 for the Apple ][ user's group meeting.
I ate at "Al's Big Burritos" for lunch!
It's not OK to burp at your grandmother.
```

```
*BuT* YoU _CaN_ Do ThIs, YoU KnOw.
We 8 eight oranges.
#
```

Output:

```
OK
suspicious
OK
suspicious
OK
suspicious
```

Added by:	Nikola P Borisov
Date:	2008-12-02
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	Mid-Central Regional ACM-ICPC Contest 2005

SPOJ Problem Set (classical)

3580. Company

Problem code: COMPANY

In Plumsoft company, there is a hierarchy among employees, i.e. some of them are bosses to the others. Person A is in charge of person B if there is a sequence of employees $P_1 = A, P_2, ..., P_k = B$, such that P_1 is P_2's boss, P_2 is P_3's boss, ..., and P_{k-1} is P_k's boss. As Plumsoft is a pretty sane company, you can assume that no two employees can be in charge of each other. The management wants to cut the costs of meetings (they eat a lot of food), so they plan to minimize the number of "A is boss of B" relations by keeping only some of the existing ones. However they want to keep all "A is in charge of B" relations. Please, help them to successfully make this transition.

Input

The first line of the input contains two integers N and M separated by a space character (1 <= N <= 1000, 1 <= M <= 10000). N is the number of employees, and M is the number of "boss" relations in the company. Employees are labeled with numbers 1 through N. Each of the next M lines contain two labels A and B separated by a space character, meaning that A is a boss of B.

Output

In the first line of the output, write a single number M_{min}, which is the minimum number of "boss" relations that the company has to keep. In the next M_{min} lines write the relations that are kept. In each line, write two labels A and B separated by a space character, meaning that A is still a boss of B. If there are multiple solutions, write any of them. Relations can be listed in any order. Each line of the output should be followed by a newline.

Example

```
Input:
5 8
3 5
1 4
4 3
1 3
4 5
1 2
1 5
2 3

Output:
5
3 5
1 4
4 3
1 2
2 3
```

Added by:	Jelani Nelson (Minilek)
Date:	2008-12-22
Time limit:	2s
Source limit:	50000B
Languages:	All
Resource:	MIT Individual Contest 2008

SPOJ Problem Set (tutorial)

3695. n-th Power

Problem code: LOGPOWER

Given an integer A, N and M, calculate $R = A^N$ *modulo M*, ie. the remainder after dividing N-th power of A by the modulus M.

Input

First line: positive integer T - numer of test cases, T<1000.
> Next T lines contain 3 integers each: A i<sub>, N_i and M_i.

> Data constraints:
> $-2^{30} < A_i < +2^{30}$

> 0 < N i<sub> < $+2^{60}$

> 2 < M i<sub> < $+2^{30}$

>

Output<h3>

For each of test cases, output the number R_i - one in each line.

Example

Input:
```
6
1 2 3
4 5 6
7 8 9
12 34 56
78 90 123
4567890 123456789012 34567890
```

Output:
```
1
4
4
16
42
781950
```

Added by:	**Robert Rychcicki**
Date:	**2009-01-10**
Time limit:	**0.400s**
Source limit:	**50000B**
Languages:	**All**

SPOJ Problem Set (tutorial)

3699. Fibonacci numbers

Problem code: LOGFIB

Let’s define:
> F(0)=0, F(1)=1.
> F(j)=F(j-1)+F(j-2), for j>1
>
> P(0)=0, P(1)=1, P(2)=2
> P(j)=P(j-1)+2P(j-3), for j>2
> Given an integer X and M, calculate the remainder of F(X) and P(X) after dividing them by the modulus M.

Input

First line: positive integer T - numer of test cases, T<20000.
> Next T lines contain 2 integers each: X i<sub>, and M_i .

> Data constraints:
> $0 < X_i < +2^{60}$
> $2 < M$ i<sub> $< +2^{30}$
>

Output<h3>

For each of test cases, output the numbers $F(X_i)$ mod M_i and $P(X_i)$ mod M_i separated by a single space - one line per test case.

Example

Input:
```
6
1 23
4 56
7 89
123 456
7890 123
123456789012 34567890
```

Output:
```
1 1
3 4
```

```
13 20
2 204
55 103
29441184 24923102
```

Added by:	**Robert Rychcicki**
Date:	**2009-01-10**
Time limit:	**15s**
Source limit:	**50000B**
Languages:	**All**

SPOJ Problem Set (tutorial)

3727. Lucky Number

Problem code: KLUCKY

The Kurukshetra OPC team observed that many online programming contests have a problem titled "Lucky Number". So we decided to have one in KOPC too.

We define the Lucky sequence as the infinite sequence of all integers, in ascending order, that can represented as any positive integer power of 5 (i.e 5^k where k is a positive integer) or as a sum of distinct positive integer powers of 5 (i.e $5^{a1} + 5^{a2} + 5^{a3} + \ldots$, where a1,a2,a3, ... are distinct positive integers). All the numbers in the lucky sequence are called lucky numbers. The first few lucky numbers are 5, 25, 30, 125, 130, 150, ...

Given n your task is to find the n^{th} lucky number.

Input

First line of input contains an integer **t**, t <= 200, representing the number of test-cases. Then t lines follow each containing one integer **n**, 1 <= n <= 8000.

Output

For each test case output the nth lucky number on a separate line. Answers will fit in a 32-bit signed integer.

Example

```
Input:
4
1
2
3
9

Output:
5
25
30
630
```

Added by: u.swarnaprakash
Date: 2009-01-17
Time limit: 2s
Source limit:50000B
Languages: All
Resource: Kurukshetra 09 OPC

SPOJ Problem Set (acm)

3903. Special Hashing

Problem code: SPHASH

Linear Probing is one of the most used Hashing techiniques. We define here a special hashing which is similar to linear probing.

The following operations are defined.

- Hash: Hash is defined for a number N to be N%k.
- Move forward : Move to the next number (the number connected by the forward link) . Initially, every number's forward link points to the itself. (currentIndex).
- Move backward : Move to the previous number (the number connected by the back link) . Initially, every number's back link points to the itself in the link (currentIndex).
- Insertion operation: Given a number N , find hash(N)=N%k where k is the size of the list. If the list[hash(N)] is empty the element is inserted at position hash(N) in the list and forward link is made to point at (currentIndex+1)%(size of list) and backward link is made to point at (currentIndex-1+sizeof list)%(size of list). If it is filled , we do move_forward/move_backward as specified and then the same process is again repeated.

  Note: Thus list is circular due to modulus property.

- Merge Operation: Let x be a index in the list which is not empty. Calculate xmin by doing a move_backward from index x till the previous index is empty . Similarily calculate xmax by doing a move_forward from index x till the next element is an empty space . Do the same for y to find out ymin and ymax. For a valid merge operation, the index x and y should not be empty and either xmax < ymin or ymax < xmin. Now, when merging x and y, if ymin > xmax, the forward link of xmax is made to point at ymin and the backward link of ymin is made to point to xmax. Same approch is applied in the other case.

  Note: for the merge operation take the min(b,c). The merge is only to be done from x(b)max to x(c)min if the merge was allowed.

Input

The first line of input contains a number representing the number of test cases. Each test case states with a line conataining two integers k(size of list) and C(operations to be applied).C lines follow. Each line contains a,b,c. a is 0 for merge operation followed by index b and c to be merged. a is 1 for insert operation and b is the element to be inserted and c is either 0 or 1(1 in case of left insertion and 0 in case right).

Output

For each operation in each test case,

Case 1: Insertion operation print the position of the hash(b). If the number cannot be inserted print the string "cannot insert element"

Case 2: Merge operation print "merge successful" if the merge was succesful and "cannot merge" if the merge operation failed.

Example

Input:

```
1
5 6
0 0 2
1 1 1
1 1 0
1 4 0
0 1 4
1 1 1
```

Output:

```
cannot merge
1
2
4
merge successful
0
```

Constraints

```
Dataset 1:T<25, k <==10000,C<=25000 Score: 100
Time limit: 5s   Memory Limit: 128MB
Dataset 2:T<8, k <==400000,C<=800000 Score: 50
Time limit: 5s   Memory Limit: 128MB
```

Added by:	Race with time
Date:	2009-02-19
Time limit:	5s
Source limit:	50000B
Languages:	All
Resource:	Code Craft 09

SPOJ Problem Set (tutorial)

3916. Bicolor

Problem code: BICOLOR

When you look at a political map of the world, each country is colored in color different from its neighbors' so that you can clearly see the borders. But as you know, the are between 192 and 195 countries in the world (depending on where you live) so it is common for two countries on the map to have the same color. After all, men can see only 16 colors ("Peach" is not a color according to me), so it has been a hard question for a long time if it is possible to color the map of the world with just 4 colors, following the rule that you are not allowed to color neighboring countries with the same color. This problem, however, is not easy at all, and we are going to simplify it a little bit. You are a Rock Star, and you are going on a tour in the galaxy. You are looking at the map of the sky and some of the stars are connected with other stars to form oddly shaped constellations. You are wondering if the stars can be bicolored (colored with just two colors) following the rule that you can not color two stars with the same color if they are directly connected with line on the map. You are bored as you are traveling towards the first star on your tour with speeds close to the speed of light so the clock in your space ship are ticking slower. Having nothing better to do, you decide to write a computer program to solve it.

Input

The input will consist of multiple maps. Each map starts with the number of stars on the map N (1 <= N <= 1024). On the next line is the number M (1<= M <=30000), the number of lines on the map connecting the stars. The stars are numbered with integers from 0 to N-1. On the next M lines you will find 2 integers - the ID-s of two stars that are connected. To denote the end of the input, the last map will have N = 0, and at this point you should stop reading.

Output

For each map in the input case, you need to output exactly one line in the output containing either the string "NOT BICOLORABLE" or "BICOLORABLE".

Example

```
Input:
3
3
0 1
1 2
2 0
5
4
0 1
0 2
0 3
0 4
0
```

Output:
```
NOT BICOLORABLE
BICOLORABLE
```

Added by:	Nikola P Borisov
Date:	2009-02-21
Time limit:	1s
Source limit:	50000B
Languages:	All

SPOJ Problem Set (tutorial)

3917. Tree Query

Problem code: TREEQ

You are given a full binary tree, and there is an integer representing the data at each node of the tree. The other members of your band are asking you questions in the following form. They give you an integer X, and you have to tell them which leaf of the tree will get X for if you follow the path from it to the root of the tree and sum the numbers in each node. If there are two such leaves, choose the one with a smaller ID. The root of the tree has an ID of 0. Its left child has an ID of 1, and its right one an ID of 2. In more mathematical form, if you have a node with an ID of x, then the left child of x is 2*x+1 and the right one is 2*x+2. The members of your band will be impressed if you answer their questions fast and correctly. Cool!

Input

The input consists of multiple test cases. Each test case starts with the number of nodes in the tree N (1<= N <= 1000). The last test case will have N = 0, so at this point you should quit. As you know a full binary tree with N nodes will have N = (2^x) - 1 nodes. The nodes are given IDs using the scheme described above, and the root has an ID of 0. On the next N lines of the input, you will find the the data that is stored in the corresponding node. On the next line after this is the integer M (1<= M <= 1000) - the number of question the band members are going to ask you. On each of the next M lines you will find one integer representing the question.

Output

For each of the questions you have to output the minimum ID of a leaf node that meets the requirements. If no leaf meats the requirements output "NOT FOUND". Refer to the example tests.

Example

```
Input:
3
1
2
3
4
0
3
4
5
0

Output:
NOT FOUND
1
2
NOT FOUND
```

Added by:	Nikola P Borisov
Date:	2009-02-21
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	Microsoft Interview

SPOJ Problem Set (tutorial)

3918. Rock Star Tour

Problem code: RSTOUR

You and your band are so famous that people from all around the world want to hear you. Even people in Bulgaria, so you decide to make a tour there. Funnily enough, all the cities in Bulgaria have names with the same length and only use lower case letters from the English alphabet. You are thinking about which cities to visit during your tour, and because you are a polite Rock Star you would like to start each concert with the sentence "Make some noise Sofia" or "Make some noise Varna". Unfortunately you are not really good with foreign languages, so you want two cities that are next to each other in the tour schedule to have names that differ in at most 2 characters. For example "aaaaa" and "aaabb" are ok to be next to each other but "aaaaa" and "cccca" are not because they have a difference of 4 characters. You are wondering between different possibilities of a start and end city of the tour. It would be useful to know for given start and end cities, what the distance of the shortest tour is. Another interesting thing is if a tour is even possible given a start and an end city.

Input

Again, the input consists of multiple test cases. For each test case, on the first line is the number N (1 <= N <= 1000) - the number of cities in Bulgaria. On the next N line you will find the name of the N cities. All names have the same length. The first integer after this is the number of possibilities you are considering - M (1<= M <= 1000). On the next M lines there will be a question in one of two possible forms: "LENGTH start-city-name end-city-name" or "POSSIBLE start-city-name end-city-name". When you are asked the LENGTH question, you have to output the shortest possible tour length (by the number of trips you will need to do, or 1 less than the number of cities) or -1 if such tour is impossible. If the question is POSSIBLE that you just need to tell if tour with this starting and ending city is possible.

Output

For each question, output a line in the form "Query #X" where x is the number of the question in the test case, and another line with the answer. If the question was LENGTH output the min length of the tour or -1 if it is impossible. For a POSSIBLE question answer with "YES" or "NO"(quotes are here for clarity only). Refer to the example tests.

Example

```
Input:
4
aaaaa
aaaab
aabbb
ccccc
4
LENGHT aaaaa aabbb
POSSIBLE aaaab ccccc
LENGHT aaaaa ccccc
POSSIBLE aaaab aaaaa
```

```
6
bbbb
abbb
aaaa
baba
aabb
aaab
5
LENGHT aabb aaaa
POSSIBLE aabb aaab
POSSIBLE aaaa bbbb
LENGHT aaab aabb
POSSIBLE aabb aaaa
```

Output:

```
Query #1
2
Query #2
NO
Query #3
-1
Query #4
YES
Query #1
1
Query #2
YES
Query #3
YES
Query #4
1
Query #5
YES
```

Added by:	Nikola P Borisov
Date:	2009-02-21
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	Google Interview

SPOJ Problem Set (tutorial)

3997. HARDWARE

Problem code: HARDWARE

Ola Clason's Hardware store is an old company where most work is done "the old way". Among other things, the company is the one and only provider of marble house numbers. These house numbers have become extremely popular among construction companies, especially the ones building luxury estates. This is of course great for Ola Clason, but also a small problem. Nisse, who has been managing the incoming orders has turned out to be a bottleneck in Ola's business. Most orders are on the form "Coconut Drive 200, 202, 204, ..., 220". This means every even number between 200 and 220. Nisse's work is to transfer an order to a list of necessary digits and other symbols. Your assignment is to write a program that automates Nisse's work with orders containing only positive integer house numbers. Nisse will still in the future process all special orders (those including non digit symbols) by hand.

Input

On the first line of input is a single positive integer n, specifying the number of orders that follow. The first line of each order contains the road name for that order. No road name is longer than 50 characters. The second line states the total number of buildings needing new marble numbers on that order. Then follows the different house number specifications on several lines. These lines are of two kinds: single number lines and multiple number lines. A single number line simply consists of the house number by itself, while a multiple number line starts with a "+"-sign, followed by three positive integer numbers: first number, last number and the interval between the house numbers. The distance between the first and last house number will always be a multiple of the house number interval. A house number will never have more than five digits. After the last house number specification line, the next order follows, if there is any.

Output

For each order, the output consists of 13 lines. The first and second lines should be identical with the first two input lines. Then, there follows 10 lines with information on how many marble digits of each kind the order consists of. These rows are on the format "Make X digit Y" where X is how many copies of digit Y they need to make. The last row states the total number Z of digits needed, on the format "In total Z digits". If there is only one digit to produce, it should say "In total 1 digit", in order to be grammatically correct.

Example

Input:

```
1
Short Street
23 addresses
+ 101 125 2
275
+ 100 900 100
```

Output:

```
Short Street
```

```
23 addresses
Make 23 digit 0
Make 22 digit 1
Make 5 digit 2
Make 4 digit 3
Make 1 digit 4
Make 5 digit 5
Make 1 digit 6
Make 4 digit 7
Make 1 digit 8
Make 3 digit 9
In total 69 digits
```

Added by:	Fabio Avellaneda
Date:	2009-03-01
Time limit:	60s
Source limit:	50000B
Languages:	All except: ICON
Resource:	I maratón interuniversitaria del circuito Redis - Acis. Sedes: Politécnico - Javeriana

SPOJ Problem Set (tutorial)

4001. JACKPOT

Problem code: JACKPOT

Bill has found the perfect way to make money playing the slot machines. After months of careful research, he has finally figured out the mechanics behind how the machines operate. Now he is ready to make profit of his findings. But first an introduction to the game. A slot machine consists of a number of wheels, usually three or four, each with a number of symbols printed on it - cherries, oranges, bells, etc. - and will show one of its symbols at a given time. To play, you insert a coin, push a button and the wheels start spinning. After spinning for a while, each wheel stops - at random it seems - at one of its symbols. If all wheels stop at the same symbol, or some nice combination of symbols, the player wins. One combination that is especially desirable is having the jackpot symbol on all wheels. This combination is simply called 'jackpot' and will make you rich for life. What Bill has discovered is that each wheel will stop at the jackpot symbol with a certain periodicity, which differs a lot between wheels. He has also figured out (after some sneeking around at the slot-machine factory) that all newly manufactured slot-machines are delivered showing the jackpot combination, and that they all have a counter at the back, telling how many times the machine has been played. This counter is always set to zero at delivery. Now, all Bill needs to do is to calculate the number of times a machine has to be played between two occurrences of the jackpot combination. We will call this number the jackpot periodicity. This is of course the same as the number of times the machine has to be played after leaving the factory, before it gives its first jackpot. Thus, with a glance at the counter on the back of a machine, Bill can figure out if it is about to give a jackpot. As Bill knows that you are a skillful computer programmer, he turns to you with the problem of calculating the jackpot periodicity. For each machine, he will give you the number of wheels, and the periodicity with which the jackpot symbol shows up on each wheel.

Input

One line with the number of machines n <= 20. For each machine, one line with the number of wheels w <= 5, and one line with w numbers, p1, ..., pw the periodicity of each wheel pk <= 1000.

Output

One line per machine: The jackpot periodicity of the machine, if it is less than or equal to a billion (10^9), otherwise output the text 'More than a billion.'.

Example

```
Input:
1
3
10 6 15

Output:
30
```

Added by:	Fabio Avellaneda
Date:	2009-03-01
Time limit:	60s
Source limit:	50000B
Languages:	All
Resource:	I maratón interuniversitaria del circuito Redis - Acis. Sedes: Politécnico - Javeriana

SPOJ Problem Set (tutorial)

4073. The 3n plus 1 problem

Problem code: PROBTNPO

Background:
Problems in Computer Science are often classified as belonging to a certain class of problems (e.g., NP, Unsolvable, Recursive). In this problem you will be analyzing a property of an algorithm whose classification is not known for all possible inputs.

The Problem:

```
Consider the following algorithm:

1. input n

2. print n

3. if n = 1 then STOP

         4. if n is odd then n = 3n + 1

         5. else n = n / 2

6. GOTO 2
```

Given the input 22, the following sequence of numbers will be printed 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

It is conjectured that the algorithm above will terminate (when a 1 is printed) for any integral input value. Despite the simplicity of the algorithm, it is unknown whether this conjecture is true. It has been verified, however, for all integers n such that 0 < n < 1,000,000 (and, in fact, for many more numbers than this.)

Given an input n, it is possible to determine the number of numbers printed (including the 1). For a given n this is called the cycle-length of n. In the example above, the cycle length of 22 is 16.

For any two numbers i and j you are to determine the maximum cycle length over all numbers between i and j.

The Input:
The input will consist of a series of pairs of integers i and j, one pair of integers per line. All integers will be less than 1,000,000 and greater than 0.

You should process all pairs of integers and for each pair determine the maximum cycle length over all integers between and including i and j.

You can assume that no operation overflows a 32-bit integer.

The Output:

For each pair of input integers i and j you should output i, j, and the maximum cycle length for integers between and including i and j. These three numbers should be separated by at least one space with all three numbers on one line and with one line of output for each line of input. The integers i and j must appear in the output in the same order in which they appeared in the input and should be followed by the maximum cycle length (on the same line).

Sample Input:

```
1 10
100 200
201 210
900 1000
```

Sample Output:

```
1 10 20
100 200 125
201 210 89
900 1000 174
```

Added by:	Coach UTN FRSF
Date:	2009-03-18
Time limit:	3s-10s
Source limit:	50000B
Languages:	All
Resource:	http://icpcres.ecs.baylor.edu/onlinejudge/

SPOJ Problem Set (tutorial)

4074. Compile Error

Problem code: CE

You are to write a program to analyse some simple arithmetical expressions. The BNF form of the defination of the expression is below.

```
<expression>::=<num><oper><num>
<num>::=0|1|2|...|99
<oper>::=+|-|*
```

Tip: You may find this problem is like the problem GALAXY very much. You may use any language you like to solve this problem.

Input

Exactly 99 test cases.

Each test case contains one line with a correct expression, without leading or trailing spaces.

Output

For each test case you should output one line contains the result of the expression without any leading zeros. You may assume this number is always a non-negative one.

Score

Score is the length of your source.

Example

```
Input:
6*7
67-25
31+11
[And 96 test cases more.]

Output:
42
42
42
[And 96 test cases more.]
```

Added by:	[Trichromatic] XilinX
Date:	2009-03-19
Time limit:	10s
Source limit:	10000B
Languages:	All
Resource:	A copy of problem GALAXY, with language restrictions and source limit modified.

SPOJ Problem Set (tutorial)

4190. A plus B

Problem code: APLUSB2

Given 2 integers, A and B, find their sum. Beware: There are no limits on A and B

Input

Two integers up to 100000 digits in length

Output

Output their sum

Example

```
Input:
1
4

Output:
5
```

Added by: Saravannan Shaan
Date: 2009-04-09
Time limit: 1s-60s
Source limit:50000B
Languages: All except: C99 strict JAVA PYTH RUBY
Resource: Classic

SPOJ Problem Set (tutorial)

4207. Mobile (Again)

Problem code: MFMOBILE

Fred is a baby. Above Fred's crib hangs a mobile. Fred is amused by this mobile. Fred has a twin sister, Mary. Above Mary's crib hangs another mobile. Fred wonders whether the mobile above his crib and the mobile above Mary's crib are the same. Help Fred.
A mobile is a collection of bars, strings, and decorative weights suspended from the ceiling. Each bar is suspended by a string tied to the exact centre of the bar. From each end of a bar hangs a string that is tied either to another bar or to a weight. The bars can rotate freely about their centres. Fred cannot tell two bars apart, even if they have different lengths. Fred also cannot tell two strings apart. Fred therefore considers two mobiles to be the same if the bars of one mobile can be rotated somehow to make the two mobiles appear identical.
Fred has even developed a notation for describing mobiles. He assigns each bar a distinct positive integer from 1 to the number of bars in the mobile, and he assigns the various objects negative integers. 1 always represents the bar suspended from the ceiling. (So, for example, a biplane might be represented by Fred as object -2, a crescent-moon might be object -57, and a star might be object -21.) Fred can only count down to -9999, so you can assume that he gave no objects lower numbers than -9999.

Input

The input contains two mobile descriptions. The first line of a mobile description contains a single nonnegative integer n (1 <= n <= 100000), indicating the number of bars in the mobile. On the next n lines, there are two numbers per line, with these two numbers representing the objects hanging from bar i.

Output

Output is composed of one line. Write "Fred and Mary have different mobiles." if Fred's information is enough to distinguish the two mobiles; otherwise, "Fred and Mary might have the same mobile.".

Example #1

```
Input:
5
2 3
4 5
-1 -2
-3 -4
-5 -6
5
2 5
-1 -2
-3 -4
-5 -6
```

```
3 4
```

Output:

```
Fred and Mary might have the same mobile.
```

Example #2

Input:

```
5
2 3
4 5
-3 -4
-1 -2
-5 -6
5
2 5
-1 -2
-3 -4
-5 -6
3 4
```

Output:

```
Fred and Mary have different mobiles.
```

Added by:	Analysis Mode (Elspeth, Knight-Errant)
Date:	2009-04-12
Time limit:	1s
Source limit:	50000B
Languages:	All except: C99 strict
Resource:	Canadian Computing Competition 2008 Stage 2 Day 1 Problem C

SPOJ Problem Set (tutorial)

4215. Inversions

Problem code: INV

Calculate the number of ways that k things can be 'chosen' from a set of n things.

Input

The first line of input is the number of tests t <= 100000. Next t lines contains two integers each n and k, separated with a single space. 0 <= k <= n <= 100000.

Output

For each test output the number of ways that k things can be 'chosen' from a set of n things modulo 1000000007.

Example

```
Input:
3
9876 5432
100 50
100000 50000

Output:
266875274
538992043
149033233
```

Added by:	Spooky
Date:	2009-04-13
Time limit:	4s
Source limit:	50000B
Languages:	All

SPOJ Problem Set (tutorial)

4245. Place the Numbers I

Problem code: PLCNUM1

Little Chucha is having a lot of fun with her new computer game. She is given a NxN board which she has to fill with the numbers 1 to N^2, no repetitions allowed. The computer calculates the sum of distances for each pair of consecutive numbers, that is, 1 -> 2, 2 -> 3, ..., N^2 -> 1. The goal is to make that sum as short as possible.

Input

Input consists of a single integer number 1<=N<=100, the size of the board.

Output

Output one possible placing of the numbers. You are to write N lines, N space separated integers each.

Example

```
Input:
3

Output:
1 2 3
4 5 6
7 8 9

Score:
Score for the example is:
Distance 1 -> 2 : 1
Distance 2 -> 3 : 1
Distance 3 -> 4 : 3
Distance 4 -> 5 : 1
Distance 5 -> 6 : 1
Distance 6 -> 7 : 3
Distance 7 -> 8 : 1
Distance 8 -> 9 : 1
Distance 9 -> 1 : 4
Sum of distances (SOD): 16, Min SOD: 10, Score: 1+16-10=7 points.
```

Added by: yandry pérez clemente
Date: 2009-04-22
Time limit: 3s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

4293. Weightlifting

Problem code: BENCH

Joe the Jock is spending the day at the gym, and he needs a certain amounts of weight for his benchpress sets.

The gym has several different types of weights, each no more than 100kg. There are an unlimited number of each type of weight.

Spare Joe the math and write a program to figure out what weights he should lift.

He thinks the bigger weights look a lot more impressive, so make sure to give him the biggest weights possible (e.g. 3,2,2 would be better than 3,2,1,1).

Unfortunately, it might also be impossible to reach the desired weight in any way.

Input

The first line is T (<50), the number of sets Joe has to do. Each of the T set descriptions are followed by a line.

The first line of each set is the total weight Joe needs, in decigrams (tenths of a gram), not more than 10^6. The second line is N (<100), the number of distinct denominations of weight. The following N lines are the weights, in decigrams.

Output

Separate the solutions for the sets by a blank line.

For each set, output one line for each type of weight he will use at least once, from heaviest to lightest. Each line will have the weight (in decigrams), a single space, and the number of that type of weight.

If a set is impossible to make, output a single line -1 instead.

Example

Input:
```
3

10
3
3
1
2

5
1
10
```

```
4799
5
2
3
6
7
42
```

Output:

```
3 3
1 1

-1

42 114
7 1
2 2
```

Added by: Paul Draper
Date: 2009-05-01
Time limit: 1s-2s
Source limit:50000B
Languages: All

SPOJ Problem Set (classical)

4323. Voting Districts

Problem code: VOTE

The land of Yu consists of N cities, labeled 1,2,...N connected by N-1 roads in such a way that there exists a path between any two cities.

For the first time in history, Yu is holding free elections. But they need to divide their new republic into voting districts.

The division of Yu into voting districts must satisfy the following:

- Each city belong to exactly one district.
- Districts must have an equal number of cities.
- Each district must be entirely self-connected (i.e. for any two cities in the same district, there exists a path between them passing only including cities of that district).

For what numbers of voting districts can this work?

Input

The first line of input is the number of test cases (less than 100) to follow.

Each test case is preceded by a blank line and begins with N (less than 10^5), the number of cities. N lines follow.

The first integer of the ith line (indexing starting at 1) is K_i, the number of cities directly connected to City i. The next K_i integers are the cities directly connected to City i.

Output

For each test case print a line of all numbers of districts into which Yu can be divided, from least to greatest, separated by spaces.

Example

`Input:``341 22 1 32 2 41 343 2 3 41 11 11 110`**`Output:`**`1 2 41 41`

Note: Notice that Yu can always be divided into 1 district (of N cities), or N districts (of 1 city each).

Added by: Paul Draper
Date: 2009-05-05
Time limit: 0.5s-8s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

4349. Power Tower

Problem code: POWTOWER

Calculate

a_1^a_2^...a_n mod b

for integers a_1,a_2,...a_n and b.

Constraints

0 < n < 10^3
0 < a_1,a_2,...a_n,b < 10^6

Input

The first line is the number of test cases to follow (less than 100), which are separated by blank lines, and each of which has the form:

```
n
a_1 a_2 ...a_n
b
```

Output

Output the answers, in order, on separate lines.

Example

```
Input:
5

2
2 3
3

3
1 1000 3142
2

1
8
1

3
12 15 2
18

4
10 10 10 34
```

```
42
```

Output:

```
2
1
0
0
4
```

Added by:	Paul Draper
Date:	2009-05-07
Time limit:	2s
Source limit:	50000B
Languages:	All

SPOJ Problem Set (classical)

4357. Enter the Matrix

Problem code: MATRIX1

Neo has to enter again in the Matrix

Neo has to enter again in the Matrix. The Matrix is represented as a two dimensional plane of coordinates x-y. When the Matrix was born, agent Smith showed up and imposed control. Unfortunately he is not alone, other agents are around him to do the same work. An agent is at some point in the Matrix given by a pair (Xi, Yi). Neo is now very furious because he cut his hair and wants to attack two agents. Neo can travel any distance to attack the first agent, but later he will be very tired, so he wants to walk the smallest distance possible to the next agent.

Neo is very busy conquering Trinity, who is a very difficult girl in terms of taking her out on a date, and he has no time to think about silly things like this, all he can think about is fun and war... Please, help him find the smallest distance possible from the first agent to the second.

Input

In the first line of the input appears one integer: T. This representes that your program must solve exactly T data sets for Neo. Each data set is not related to the previous one. In the first line of each data set appears one integer: N, representing that there are N agents in that layout of agents. Following this line, N pairs of integers will appear, each pair in a new line, representing the position of each agent.

Output

For each data set your program must output only one line represents the smallest distance between two agents. Please for avoid decimal roundings write the result whit zero decimal places after the comma.

Example

Input:

2

4

1 1

-1 3

4 4

2 2

3

1 1

2 4

5 1

Output:

1

3

Constraints

- 1 <= T <= 20

- 2 <= N <= 100000

- -1000000 <= Xi, Yi <= 1000000

Added by: Reinier César Mujica Hdez
Date: 2009-05-13
Time limit: 2s
Source limit:50000B
Languages: All except: SCALA JS
Resource: My own problem

SPOJ Problem Set ()

4375. Bicoloring II

Problem code: BICOII

In 1976 the "Four Color Map Theorem" was proven with the assistance of a computer. This theorem states that every map can be colored using only four colors, in such a way that no region is colored using the same color as a neighbor region. Here you are asked to solve a simpler similar problem. You have to decide whether a given arbitrary connected graph can be bicolored. That is, if one can assign colors (from a palette of two) to the nodes in such a way that no two adjacent nodes have the same color. You can assume:

- No node will have an edge to itself.
- The graph is non-directed. That is, if a node a is said to be connected to a node b, then you must assume that b is connected to a.

Input

The input consists of several test cases. Each test case starts with a line containing the number n (1 < n < 200) of different nodes. The second line contains the number of edges l. After this, l lines will follow, each containing two numbers that specify an edge between the two nodes that they represent. A node in the graph will be labeled using a number a (0 <= a <= n). An input with n = 0 will mark the end of the input and is not to be processed.

Output

You have to decide whether the input graph can be bicolored or not, and print it as shown below.

Example

```
Input:
3
3
0 1
1 2
2 0
9
8
0 1
0 2
0 3
0 4
0 5
0 6
0 7
0 8
0

Output:
NOT BICOLORABLE.
BICOLORABLE.
```

Added by:	Daniel Gómez Didier
Date:	2009-05-14
Time limit:	2s
Source limit:	50000B
Languages:	All
Resource:	ACM javeriana - acm.javeriana.edu.co

SPOJ Problem Set (classical)

4460. Tic Tac Toe - Best Move

Problem code: TITATO

Tic-tac-toe is played on a 3x3 grid between two players. All squares begin empty, and play is as follows:

1. The first player places an X on any empty square.
2. The second player then places an O on any emtpy square.
3. The first player then places another X onany empty square
4. ...

The players continue in this matter until one of the following occurs:

- Three X's form a straight line (horizontal, vertical, or diagonal), in which case the first player wins.
- Three O's form a straight line, in which case the second player wins.
- The grid is filled with X's and O's (with no three X's or three O's in a line), in which case the players tie.

Le the tic-tac-toe grid be numbered as follows:

1	2	3
4	5	6
7	8	9

Write a computer program that will decide the best move for the current player based on the previous moves of the two players. In case of multiple best moves, choose the one with the lowest number in the grid above.

The "best" move is defined as the move that guarantees the highest outcome (win>tie>loss). An outcome is "guaranteed" if the player can always cause it to happen, regardless of the other's player's moves.

All tic-tac-toe games given will still be in progress, i.e. there is an empty space and no player has won yet.

Input

The first line is the number of test cases (fewer than 2000) to follow.

Each test case consists of two lines. The first line is the number of moves that have already happened. The second line lists the grid numbers of the previous moves, in order, each followed by space.

Output

For each test case, output the best move with the lowest number for the current player.

Example

Input:421 2 21 5 0 81 2 3 4 5 8 6 9 **Output:**4217

Added by: Paul Draper
Date: 2009-06-01
Time limit: 1s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

4474. Longest Palindromic Substring

Problem code: LPS

A palindrome is a string that is the same as its reverse. Example "malayalam", "dad", "appa" etc. In this problem you are asked to find the length of the longest contiguous substring of a given string, which is a palindrome.

Input

The first line consists of a single integer N, the no. of characters in the string. 1 <= N <= 100000.

Second line is a string with N characters, where the characters are always lowercase english alphabets, ie 'a' to 'z'.

Output

A single line with an integer representing the length of longest palindromic substring.

Example

`Input:``5ababa`

`Output:``5`

Added by: Srivatsan B
Date: 2009-06-04
Time limit: 1s-5s
Source limit:50000B
Languages: All
Resource: http://opc.iarcs.org.in/problems/iarcs-feb05-ad-1

SPOJ Problem Set (classical)

4492. Magic1

Problem code: MAGIC1

A Magic Square is an N by N matrix such that the sum of all the elements in each row, each column and the two diagonals is the same. Also, all the elements of a Magic Square are different and are in the range from 1 to N^2.

Your are given a matrix with some element missing. Your task is to complete the matrix so that it is a Magic Square (if it is possible).

Input

The first line of the standard input will contain one integer **N**, (1 <= **N** <= 5). Each of the next **N** lines will contain **N** integers representing the elements in the matrix. The number 0 represents an empty element.

Output

If it is possible to complete the matrix so that it is a Magic Square, then output the matrix to the standard output - in **N** lines output **N** integers separated by a single space that represent the matrix. If it is impossible to fill out the empty elements so that the marix is a Magic Square, then output -1 to the standard output.

Example

`Input:`

2
1 0
2 3

`Output:-1`

Added by: Tomas. Bob
Date: 2009-06-12
Time limit: 0.400s-1s
Source limit:50000B
Languages: All
Resource: www.z-trening.com

SPOJ Problem Set (classical)

4493. Equation

Problem code: EQUAD1

You are given an equation, which when you solve, you get a huge prize.

The equation consists out of integers between 1 and 1000, signs **+**, **-** (plus and minus) and parenthesis **(** and **)**, and finally one **?** which represents the unknown variable. The question mark can be on either side of the equation.

Input

From the first line of the standard input read the equation, the equation will contain at most 1000000 characters.

Output

You shoud write the wanted number.

Example

`Input:5 + ? = 32 - 2Output:`

25

Added by: Tomas. Bob
Date: 2009-06-12
Time limit: 0.009s
Source limit:50000B
Languages: All
Resource: www.z-trening.com

SPOJ Problem Set (tutorial)

4500. Helms Deep

Problem code: LOTR1

We have all heard about good triumphing over evil. A perfect example would be the Battle of Helms Deep.

Lets probe a little deeper...

The people (especially warriors) of Rohan (good side) could not have known that they would win the battle before it was actually fought. This would have forced them to think up of some other alternate means of escape. A river flows near the fortress and naturally, they think this to be the best escape route. They build the biggest ship possible with the available materials.

This ship can carry a total weight of W($0<W<10^9$) kilograms. There are a total of n ($0<n<10000$) people at Helms Deep having weights wi ($0<wi<10^9$).

What is the maximum number of people who can escape?

Input

1st line contains t, the number of testcases. First line of each test case contains the number of people n($0<n<10000$) and the capacity of the ship W($0<W<10^9$). The next n lines each contain the weights wi($0<wi<10^9$) of the i th person. All numbers in the input file are integers.

Output

One line giving the maximum number of people who can escape.

Example

`Input:``13 3213`**`Output:`**`2`

Added by: pradeep
Date: 2009-06-15
Time limit: 1s
Source limit:50000B
Languages: All
Resource: Lord of the Rings (2 towers)

SPOJ Problem Set (tutorial)

4522. Adding Base36 BigNumbers

Problem code: UCI2009A

Ivan Ivanovich, the evil problemsetter, got tired of you and your team always using Java's BigInteger class. In an attempt to get back at you, he created this problem. You are to write a program which calculates the sum of pairs of integers. Simple, isn't it?

Not so fast, coder. You will be dealing with base36 integers. Haven't you heard of them? It's easy.

- Digits from '0' to '9' represent the values 0 to 9.
- Digits from 'A' to 'Z' represent the values 10 to 35.

So ACMICPC_{36} is 22531225296_{10} and UCI2009_{36} is 66059390601_{10}.

Input

Input starts with an integer T, representing the number of test cases (1<=T<=100). For each test case you will be given two base36 positive integers X1 and X2, whose lengths won't exceed 1000 digits.

Output

Print T sums, no leading zeroes, one line each.

Example

Input:2ACMICPCUCI2009ACMICPCUCI2009**Output:**14P4KCPL14P4KCPL

Added by: yandry pérez clemente
Date: 2009-06-23
Time limit: 3s
Source limit:50000B
Languages: All

SPOJ Problem Set (classical)

4543. Mysterio s Menace

Problem code: SPIDEY1

The nefarious Mysterio an expertise in illusions has a mind-warping hold on the City of New York, and it's up to Spidey to stop him before he takes over the entire city! Mysterio uses tiny cubes to create holograms which makes up his illusions. Mysterio has disguised the whole New York City into his own dream world. Mysterio now has created this illusive world with hologram cubes on a linear arena. To break this illusion Spiderman has to pick up as many cubes as he can from the arena. But wait thats not so easy!! Mysterio has disguised the arena too!!
The arena consists of rectangular rocks (of zero width) whose starting and ending x-coordinates are given.(The y-coordinates of the rocks is immaterial). For example consider the configuration of the rocks as follows: 2-4 ,3-8 ,4-8 ,8-9 ,9-10 here the first rock spans between x coordinate 2 to 4 and so on. Now if Spidey steps on a rock all other rocks which have overlapping segments with it disappear ie.,if he steps on rock 1 then rock 2 disappears as there is an overlapping segment namely the segment 3-4 but when he steps on rock 8-9 no other rock disappears as this rock has no overlapping segment with any other rock.Assume that Spidey can jump from any rock to any other rock and if he lands on a rock he destroys the hologram cube present on the rock. Poor Peter Parker is out of mind in this menace! Help him find the maximum number of cubes he can destroy.
Input Format
The first line of the input consists of a single integer T(1<=T<=100) specifying the number of test cases to follow .The first line of each test case is a single integer N(2<=N<=100000) the number of rocks in the arena. The next N lines of each test case consist of two space separated integers X1 X2 specifying the starting X-coordinate and the ending X-coordinate of rocks. The i+1 th line of each test case specifies the configuration of the i th rock.
0 <= X1 ,X2 <= 1000000000
Output Format
For each test case output a single integer M the maximum number of cubes that Spiderman can destroy.
SAMPLE INPUT:
2
5
2 4
3 8
4 8
8 9
9 10
7
26 29
23 27
25 28
30 32
32 37
27 31

31 35
SAMPLE OUTPUT:
4
3 The nefarious Mysterio an expertise in illusions has a mind-warping hold on the City of New York, and it's up to Spidey to stop him before he takes over the entire city! Mysterio uses tiny cubes to create holograms which makes up his illusions. Mysterio has disguised the whole New York City into his own dream world. Mysterio now has created this illusive world with hologram cubes on a linear arena. To break this illusion Spiderman has to pick up as many cubes as he can from the arena. But wait thats not so easy!! Mysterio has disguised the arena too!!

The nefarious Mysterio an expertise in illusions has a mind-warping hold on the City of New York, and it's up to Spidey to stop him before he takes over the entire city! Mysterio uses tiny cubes to create holograms which makes up his illusions. Mysterio has disguised the whole New York City into his own dream world. Mysterio now has created this illusive world with hologram cubes on a linear arena. To break this illusion Spiderman has to destroy as many cubes as he can from the arena. But wait thats not so easy!! Mysterio has disguised the arena too!!

The arena consists of rectangular rocks (of zero width) whose starting and ending x-coordinates are given.(The y-coordinates of the rocks is immaterial). For example consider the configuration of the rocks as follows: 2-4 ,3-8 ,4-8 ,8-9 ,9-10 here the first rock spans between x coordinate 2 to 4 and so on. Now if Spidey steps on a rock all other rocks which have overlapping segments with it disappear ie.,if he steps on rock 1 then rock 2 disappears as there is an overlapping segment namely the segment 3-4 but when he steps on rock 8-9 no other rock disappears as this rock has no overlapping segment with any other rock.Assume that Spidey can jump from any rock to any other rock and if he lands on a rock he destroys the hologram cube present on the rock. Poor Peter Parker is out of mind in this menace! Help him find the maximum number of cubes he can destroy.

Input Format

The first line of the input consists of a single integer T(1<=T<=100) specifying the number of test cases to follow .The first line of each test case is a single integer N(2<=N<=100000) the number of rocks in the arena. The next N lines of each test case consist of two space separated integers X1 X2 specifying the starting X-coordinate and the ending X-coordinate of rocks. The i+1 th line of each test case specifies the configuration of the i th rock.

0 <= X1 ,X2 <= 1000000000

Output Format

For each test case output a single integer M the maximum number of cubes that Spiderman can destroy.

SAMPLE INPUT:

2
5
2 4
3 8
4 8
8 9
9 10
7
26 29
23 27
25 28
30 32
32 37
27 31
31 35

SAMPLE OUTPUT:

4
3

Added by:	Rajeev Kumar.J
Date:	2009-07-01
Time limit:	3s
Source limit:	50000B
Languages:	All

SPOJ Problem Set (acm)

4566. Đe^’m hinh chu+~ nhât

Problem code: LQDRECT

Cho môt ba?ng kích thu+o+’c MxN, đu+o+.c chia thanh lu+o+’i ô vuông đo+n vi. M dong N côt. (1<=M<=1000;1<=N<=300)
Tren các ô cu?a ba?ng ghi sô 0 hoăc 1. Các dong cu?a ba?ng đu+o+.c đánh sô 1, 2... M theo thu+’ tu+. tu+‘ tren xuông du+o+’i va các côt cu?a ba?ng đu+o+.c đánh sô 1, 2..., N theo thu+’ tu+. tu+‘ trái qua pha?i
Yeu câu:
Đe^’m sô hinh chu+~ nhât gôm các ô cu?a ba?ng thoa? man các đie^‘u kie^.n sau:
1 - Hinh chu+~ nhât đó có 4 ô o+? 4 đi?nh la 4 ô khác nhau
2- 4 ô o+? đi?nh đe^‘u la sô 1
3- Ca.nh hinh chu+~ nhât song song vo+’i ca.nh ba?ng

Input

Dong 1: Ghi hai sô M, N
M dong tie^’p theo, dong thu+’ i ghi N sô ma sô thu+’ j la sô ghi tren ô (i, j) cu?a ba?ng

Output

Gôm 1 dong duy nhât ghi sô hinh chu+~ nhât tho?a man yeu câu .
Example

```
Input:4 41 0 0 10 1 1 11 1 1 11 1 1 1Output:14Ke^’t qua? trong pha.m vi int64 cu?a pascal va long long cu?a C,C++
```

Added by: Ruan Shee Zhong
Date: 2009-07-05
Time limit: 1s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

4762. Desde Hasta

Problem code: DESDEHAS

Mo and Larry have devised a way of encrypting messages. They first decide secretly on the number of columns and write the message (letters only) down the columns, padding with extra random letters so as to make a rectangular array of letters. For example, if the message is "There's no place like home on a snowy night" and there are five columns, Mo would write down

```
t o i o yh p k n ne l e a ir a h s ge c o n hs e m o tn l e w x
```

Note that Mo includes only letters and writes them all in lower case. In this example, Mo used the character 'x' to pad the message out to make a rectangle, although he could have used any letter. Mo then sends the message to Larry by writing the letters in each row, alternating left-to-right and right-to-left. So, the above would be encrypted as

```
toioynnkpheleaigshareconhtomesnlewx
```

Your job is to recover for Larry the original message (along with any extra padding letters) from the encrypted one.

Input

There will be multiple input sets. Input for each set will consist of two lines. The first line will contain an integer in the range 2...20 indicating the number of columns used. The next line is a string of up to 200 lower case letters. The last input set is followed by a line containing a single 0, indicating end of input.

Output

Each input set should generate one line of output, giving the original plaintext message, with no spaces.

Example

Input:5toioynnkpheleaigshareconhtomesnlewx3ttyohhieneesiaabss0

Output:

```
theresnoplacelikehomeonasnowynightxthisistheeasyoneab
```

Added by:	Coach UTN FRSF
Date:	2009-09-02
Time limit:	1s-2s
Source limit:	50000B
Languages:	All
Resource:	Original version TO and Fro here in spoj

SPOJ Problem Set (tutorial)

4765. The 3n plus 1 problem V2

Problem code: PROBTRES

Background:
Problems in Computer Science are often classified as belonging to a certain class of problems (e.g., NP, Unsolvable, Recursive). In this problem you will be analyzing a property of an algorithm whose classification is not known for all possible inputs.

The Problem:

```
Consider the following algorithm:1. input n2. print n3. if n = 1 then STOP        4. if n is odd then n = 3n + 1        5. else n = n / 26. GOTO 2
```

Given the input 22, the following sequence of numbers will be printed 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

It is conjectured that the algorithm above will terminate (when a 1 is printed) for any integral input value. Despite the simplicity of the algorithm, it is unknown whether this conjecture is true. It has been verified, however, for all integers n such that 0 < n < 1,000,000 (and, in fact, for many more numbers than this.)

Given an input n, it is possible to determine the number of numbers printed (including the 1). For a given n this is called the cycle-length of n. In the example above, the cycle length of 22 is 16.

For any two numbers i and j you are to determine the maximum cycle length over all numbers between i and j.

The Input:
The input will consist of a series of pairs of integers i and j, one pair of integers per line. All integers will be less than 1,000,000 and greater than 0.

You should process all pairs of integers and for each pair determine the maximum cycle length over all integers between and including i and j.

You can assume that no operation overflows a 32-bit integer.

The Output:
For each pair of input integers i and j you should output i, j, and the maximum cycle length for integers between and including i and j. These three numbers should be separated by at least one space with all three numbers on one line and with one line of output for each line of input. The integers i and j must appear in the output in the same order in which they appeared in the input and should be followed by the maximum cycle length (on the same line).

Sample Input:`1 10100 200201 210900 1000`**Sample Output:**`1 10 20100 200 125201 210 89900 1000 174`

Added by:	Coach UTN FRSF
Date:	2009-09-02
Time limit:	1s-10s
Source limit:	50000B
Languages:	All except: ERL JS PERL 6

SPOJ Problem Set (tutorial)

4768. Cool Numbers

Problem code: COOLNUM

Eric likes interesting numbers like 64. It turns out that 64 is both a square and a cube, since 64 = 8^2 and 64 = 4^3 . Eric calls these numbers cool.

Write a program that helps Eric figure out how many integers in a given range are cool.

Input

On the first line of input, you are given an integer a such that a >= 1 and a <= 10^8 . On the second line of input, you are given an integer b such that a <= b and b <= 10^8 .

Output

The output should be the number of cool numbers in the range a to b (inclusively: that is, a and b would count as cool numbers in the range if they were actually cool).

Example

```
Input:
1
100

Output:
2

Input:
100
1000

Output:
1
```

Added by: Mislav Balunović
Date: 2009-09-02
Time limit: 0.100s
Source limit:50000B
Languages: All
Resource: Canadian Computing Competition

SPOJ Problem Set (tutorial)

4770. Add and Reverse Sums

Problem code: ADDREVUT

The Problem

The "reverse and add" method is simple: choose a number, reverse its digits and add it to the original. If the sum is not a palindrome (which means, it is not the same number from left to right and right to left), repeat this procedure.

For example:
195 Initial number
591

786
687

1473
3741

5214
4125

9339 Resulting palindrome

In this particular case the palindrome 9339 appeared after the 4th addition. This method leads to palindromes in a few step for almost all of the integers. But there are interesting exceptions. 196 is the first number for which no palindrome has been found. It is not proven though, that there is no such a palindrome.

Task :
You must write a program that give the resulting palindrome and the number of iterations (additions) to compute the palindrome.

You might assume that all tests data on this problem:
- will have an answer ,
- will be computable with less than 1000 iterations (additions),
- will yield a palindrome that is not greater than 4,294,967,295.

The Input

The first line will have a number N with the number of test cases, the next N lines will have a number P to compute its palindrome.

The Output

For each of the N tests you will have to write a line with the following data : minimum number of iterations (additions) to get to the palindrome and the resulting palindrome itself separated by one space.

Sample Input

```
3195265750
```

Sample Output

4 9339
5 45254
3 6666

Added by:	Coach UTN FRSF
Date:	2009-09-03
Time limit:	2s-3s
Source limit:	50000B
Languages:	All
Resource:	http://online-judge.uva.es/p/v100/10018.html

SPOJ Problem Set (main)

4786. Invertir Cadena

Problem code: PC1

Elabore un programa que dado un vector de letras, lo invierta y lo imprima.

El vector debe de manejarse con memoria dinámica y aritmética de apuntadores y la función realizada debe de ser recursiva.
Además, el programa debe ir divido por funciones y en comentario la especificación formal de cada una.

Input

La entrada comienza con un número t indicando los casos de prueba. En cada caso hay un número x, siendo x el tamano del vector, seguido por x líneas donde en cada una de ellas se encuentra un caracter del vector.

Output

Para cada caso de prueba imprimir el vector invertido seguido por un salto de línea.

Example

`Input:``24hola2ma`**`Output:`**`aloham`

Added by: Fabio Avellaneda
Date: 2009-09-07
Time limit: 1s
Source limit:50000B
Languages: C++

SPOJ Problem Set (main)

4787. Vocales en un vector de letras

Problem code: PC2

Elabore un programa que dado un vector de letras, cuente e imprima el número de vocales que hay en él.
El vector debe de manejarse con memoria dinámica y aritmética de apuntadores y la función realizada debe de ser recursiva.
Además, el programa debe ir divido por funciones y en comentario la especificación formal de cada una.

Input

La entrada comienza con un número t indicando los casos de prueba. En cada caso hay un número x, siendo x el tamano del vector, seguido por x líneas donde en cada una de ellas se encuentra un caracter del vector.

Output

Para cada caso de prueba imprimir el número de vocales contenidas en el vector de letras.

Example

`Input:``24hola2ma`**`Output:`**`21`

Added by: Fabio Avellaneda
Date: 2009-09-07
Time limit: 1s
Source limit:50000B
Languages: C++

SPOJ Problem Set (tutorial)

4789. De decimal a binario

Problem code: PC4

Elabore una función que convierta un número en base decimal a base binaria. Dicha función deberá ser recursiva.
El programa debe ir divido por funciones y en comentario la especificación formal de cada una.

Input

La entrada consiste en un número t indicando los casos de prueba. Luego vienen t lineas, cada una con el número a convertir.

Output

Para cada caso de prueba imprimir el número ingresado en base binaria.

Example

```
Input:376131Output:
11111010000011
```

Added by: Fabio Avellaneda
Date: 2009-09-07
Time limit: 1s
Source limit:50000B
Languages: C++

SPOJ Problem Set (classical)

4826. JOURNEY

Problem code: TPJOUR

Cho 1 đô thi. có N đi?nh, M ca.nh 2 chie^‘u. 2 đi?nh đu+o+.c nôi vo+’i nhau nhie^‘u nhât la 1 ca.nh.

1 đu+o+‘ng đi la 1 chuôi các đi?nh sao cho các đi?nh ke^‘ nhau có ca.nh nôi vo+’i nhau.

1 đu+o+‘ng đi đe.p pha?i tho?a man :

- Chuôi đi?nh có đô dai la D.

- Các đi?nh tu+‘ 1 đe^’n K pha?i xuât hie^.n ít nhât 1 lân.

Yeu câu : Tính sô đu+o+‘ng đi đe.p mod 1000000009.

Gio+’i ha.n N<=20, D<=1000000000. K<=7.

Input

- 4 sô N,M,K,D

- M dong môi dong 2 sô u,v cho bie^’t có đu+o+‘ng đi 2 chie^‘u ca.nh nôi 2 đi?nh u va v

Output

- Sô đu+o+‘ng đi đe.p mod 1000000009.

Sample input

4 4 2 3

1 2

2 3

3 1

2 4

Sample output

10

#Gia?i thích

Các đu+o+‘ng đi có the^? la :

- 1_2_1
- 1_2_3
- 1_2_4
- 1_3_2
- 2_1_2
- 2_1_3
- 2_3_1
- 3_1_2
- 3_2_1
- 4_2_1

Added by: dqd
Date: 2009-09-18
Time limit: 1s-5s
Source limit:50000B
Languages: All except: ERL TECS JS
Resource: Su+u tâm

SPOJ Problem Set (classical)

4830. Z meet

Problem code: KZWC

The Z-meet is an annual alumni meet of the department of Computer Science at PSG college of technology,where students of different batches
meet.The batch names are named in a alphabetical order and ending with Z(hence the name Z-meet).That is Aztecz,Byzandierz,Calitz,Dextroblitz,Espritz,
F5erz and Griffinxeritz.The event is held in a hall and the students are seated in random order.The event co-ordinators hav a problem.They need to find
the strength of attendence of the event by different batches which is calculated as follows:
1.each person in the hall shouts out a number which is calculated as (year of Z-meet - year of pass-out)^2 if he is a alumni or
(year of Z-meet - year of entering)^2 if he is still a student.assume the batch of the year in which Z-meet is not passedout.
2.the numbers shouted by each person in the hall is added to get the strength of attendence.
The year of passing & year of entering of the different batches are given:
Aztecz 2001-2005
Byzandiarz 2002-2006
Calitz 2003-2007
Dextroblitz 2004-2008
Espritz 2005-2009
F5erz 2006-2010
Griffinxeritz 2007-2011
for example if the event is held in 2008 the Aztecz batch member shouts (2008-2005)^2=9
and a Dextroblitz member shouts (2008-2004)^2=16
help the event co-ordinator by writing a program to find strength of attendence.

Input

ther is a single positive integer T on the first line indicating the number of test cases to follow.
then there are T lines each containing a year in which Z-meet is held and a string giving the
seating arrangement of the alumni and students.that is A refers to Aztecz member C refers to Calitz member
and so on.

Output

For every string,output a single line containing the strength of attendence of the event.

Example

Input:32008 ADECBA2006 BABACAD2009 BEGFADEG**Output:**484875

Added by:	balaji
Date:	2009-09-19
Time limit:	4s
Source limit:	50000B
Languages:	All except: TCL SCALA
Resource:	kruzade 09 practice

SPOJ Problem Set (tutorial)

4832. Pitagoras y Fermat

Problem code: PITYFERM

FERMAT vs. PITAGORAS

Computer generated and assisted proofs and verification occupy a small niche in the realm of Computer Science. The first proof of the four-color problem was completed with the assistance of a computer program and current efforts in verification have succeeded in verifying the translation of high-level code down to the chip level.

This problem deals with computing quantities relating to part of Fermat’s Last Theorem: that there are no integer solutions of

Given a positive integer *N*, you are to write a program that computes two quantities regarding the solution of

where *x*, *y*, and *z* are constrained to be positive integers less than or equal to *N*. You are to compute the number of triples (*x*,*y*,*z*) such that $x<y<z$, and they are relatively prime, i.e., have no common divisor larger than 1. You are also to compute the number of values $0 < p < N$ such that *p* is not part of any triple (not just relatively prime triples).

Input

The input consists of a sequence of positive integers, one per line. The first one is the number of test cases. Each following integer in the input file represents N and will be less than or equal to 1,000,000.

Output

For each integer *N* in the input file print two integers separated by a space. The first integer is the number of relatively prime triples (such that each component of the triple is <= N). The second number is the number of positive integers <= N that are not part of any triple whose components are all <=N . There should be one output line for each input line.

Example Input

```
31025100
```

Example Output:

```
1 44 916 27
```

Added by: Coach UTN FRSF
Date: 2009-09-19
Time limit: 2s-5s
Source limit:50000B
Languages: All except: SCALA
Resource: uva

SPOJ Problem Set (classical)

4834. Happy Sequence

Problem code: KZHAPPY

The Kruzade OPC team felt we should have a happy ending to the Kruzade online coding event.We define the happy sequence as follows:
let the sum of the squares of the digits of a positive integer s0 be represented by s1. In a similar way, let the sum of the squares of the digits of s1 be
represented by s2, and so on.If si=1 for some i>=1, then the original integer s0 is said to be happy. For example, starting with 7 gives the sequence 7, 49,
97, 130, 10, 1, so 7 is a happy number. The first few happy numbers are 1, 7, 10, 13, 19, 23, 28, 31, 32, 44, 49...
You have been hired to find out the nth happy number in the sequence.

Input

First line contains an integer T, representing the number of test-cases. Then T lines
follow each containing one integer n, 1 <= n <= 500.

Output

For each test case output on a line the nth happy number in the sequence.

Example

Input:

```
310520
```
Output:
```
4419100
```

Added by: balaji
Date: 2009-09-19
Time limit: 4s
Source limit:50000B
Languages: All except: TCL SCALA
Resource: kruzade 09 practice

SPOJ Problem Set (classical)

4835. Weightlift challenge

Problem code: KZGREEDY

In a weightlifting challenge between Mr.SixPack and Mr.EightPack,the weights are arranged on a bar in random order. On each weight, is
written a positive integer denoting the weight in kg. Players take turns removing a weight from either end of the bar and placing the weight in his
bar(Mr.SixPack starts the game). The player whose bar add up to the highest weight wins the challenge(assuming he is strong enough to lift it). Now
one strategy is to simply pick the weight at the end that is the largest -- we'll call this the greedy strategy.However, this is not always optimal,
as the following example shows: (Mr.SixPack would win if he would first pick the 3kg instead of the 4kg).
3 2 10 4
You are to determine exactly how bad the greedy strategy is for different challenges when Mr.EightPack uses it but Mr.SixPack is free to use any
strategy he wishes.

Input

There will be multiple test cases. Each test case will be contained on one line. Each line will start with an even integer n followed by n positive integers
denoting the weights arranged on the bar.A value of n = 0 indicates end of input. You may assume that n is no more than 100. Furthermore, you may
assume that the sum of the weights in the list does not exceed 10,000.

Output

For each test case you should print one line of output of the form:
In challenge m, the greedy strategy might lose by as many as p kilograms.
where m is the number of the challenge (starting at challenge 1) and p is the maximum possible difference between weight of Mr.SixPack's bar and
weight of Mr.EightPack's bar when Mr.EightPack uses the greedy strategy. When employing the greedy strategy, always take the larger end.
If there is a tie, remove the left end.

Example

```
Input:4 3 2 10 48 1 2 3 4 5 6 7 88 2 2 1 5 3 8 7 30Output:
In challenge 1, the greedy strategy might lose by as many as 7 kilograms.In challenge 2, the greedy strategy might lose by as many as 4 kilograms.In challenge 3, the greedy strategy might lose by as many as 5 kilograms.
```

Added by:	balaji
Date:	2009-09-19
Time limit:	1s
Source limit:	50000B
Languages:	All except: TCL SCALA
Resource:	kruzade 09 practice

SPOJ Problem Set (main)

4872. Good Sequence

Problem code: KZGD

It is good to hav an auspicious start to any event.The kruzade OPC team felt that online coding event should
also have an auspicious start.As a mark of auspiciousness, we define good sequence as follows:
A good number is defined as a non-negative number that has an odd number of 1s in its binary expansion(that is when the decimal number is converted to base 2).
for eg.
1=1 num of 1s in binary equiv=1(odd) so,1 is a good number
2=10 num of 1s in binary equiv=1(odd) so,2 is a good number
3=11 num of 1s in binary equiv=2(even) so,3 is not a good number

The good sequence is the collection of good numbers.
The good sequence goes like this:
1,2,4,7,8,11,13,14,16,19...

You have been hired to find out the nth good number in the sequence.

Input

First line contains an integer T, representing the number of test-cases. Then T lines follow each containing one integer n, 1 <= n <= 500.

Output

For each test case output on a line the nth good number in the sequence.

Example

```
Input:
310520

Output:
19838
```

Added by: balaji
Date: 2009-09-28
Time limit: 4s
Source limit:50000B
Languages: All
Resource: kruzade 09 main

SPOJ Problem Set (main)

4873. ABC Blocks

Problem code: KZBLK

ABC college of technology has a number of blocks which houses a number of academic departments and other facilities.
The following are some of the blocks and the facilities there.

1 a block-admin
2 b block-book depot
3 d block-conf hall
4 f block-canteen
5 i block-industry
6 m block-applied science block
7 n block-management
8 o block-hostel
9 j block-mech block
10 t block-textile block
11 eb bock-computer science block

Each block is denote by a number.Some of these blocks are linked to each other through bridges,to navigate easier.
You are to guide a student from a block to the destination through all possible paths.

Input

The first line consists of a single integer which is the destination.
The following lines each consist of a pair of positive
integers separated by a space and terminated by a new-line. They represent the blocks connected by a bridge.
For example, if 1 2 appears on a line, then there is a bridge between block a and b. The final line consists of a pair of
0's.

Output

Your output must consist of a line for each valid route from the a block to the destination.The blocks must
appear separated by a space, terminated by a new-line. Include only routes which do not pass through any
blocks more than once.

Example

Input:
```
51 22 31 44 52 53 40 0
```
Output:
```
1 2 3 4 51 2 51 4 3 2 51 4 5
```

Added by:	balaji
Date:	2009-09-28
Time limit:	4s
Source limit:	50000B
Languages:	All
Resource:	kruzade 09 mains

SPOJ Problem Set (classical)

4874. KZGAME

Problem code: KZGM

Esoteric Inc. is a social gaming company that needs to create an aplication to be run on a social networking site.The game resembles Scramble,a popular word game to enrich your vocabulary.The game comes with a set of words,each word written on a piece of plastic.
Users challenge each other by picking two letters (let's call them C1 and C2) and then trying to connect these letters by finding a sequence of one or more words (we'll refer to them as W1, W2,..., Wn) where the first word W1 starts with C1 and the last word Wn ends with C2 .
Each two consecutive words in the sequence (Wi,Wi+1) must overlap with at least two letters. Word X overlaps by k letters with word Y if the last k letters of X are the same as the first k letters of Y .

scramble

Take for example
the figure, where 'a' was connected to 's'
using the two-word sequence ‘‘against" and ‘‘students".

You have been hired to write a program that takes a dictionary of words and determines the winning sequence connecting two given letters.

Input

The first line of input consists of an integer n followed by n lines each containing a word for the dictionary.The next two lines should accept the start letter and the end letter of the word sequence.

Output

Output should contain a series of words which form a sequence starting and ending with the letters given in the input.There can be more than one sequence.If output contains many sequences,the sequences should be printed in lexicographic order.

Example

Input:

```
5skykyteskypepenentersr
```

Output:

```
skype pen enter
```

Added by:	balaji
Date:	2009-09-28
Time limit:	1s
Source limit:	50000B
Languages:	All except: SCALA
Resource:	kruzade 09 mains

SPOJ Problem Set (main)

4875. Scramble game

Problem code: KZGME

Esoteric Inc. is a social gaming company that needs to create an aplication to be run on a social networking
site.The game resembles Scramble,a popular word game to enrich your vocabulary.The game comes with a set of
words,each word written on a piece of plastic. Users challenge each other by picking two letters
(let’s call them C1 and C2) and then trying to connect these letters by finding a sequence of one or more words
(we’ll refer to them as W1, W2,..., Wn) where the first word W1 starts with C1 and the last word Wn ends with C2 .
Each two consecutive words in the sequence (Wi,Wi+1)must overlap with at least two letters. Word X overlaps
by k letters with word Y if the last k letters of X are the same as the first k letters of Y . Take for example
the figure, where ‘a’ was connected to ‘s’ using the two-word sequence ‘‘against" and ‘‘students".

Scramble

You have been hired to write a program that takes adictionary of words and determines the winning sequence connecting two given letters.

Input

The first line of input consists of an integer n followed by n lines each containing a word for the dictionary.The next two lines should accept the start letter and the end letter of the word sequence.

Output

Output should contain a series of words which form a sequence starting and ending with the letters given in the input.There can be more than one sequence.If output contains many sequences,the sequences
should be printed in lexicographic order.

Example

```
Input:

5skykyteskypepenentersr
Output:
skype pen enter
```

Added by:	balaji
Date:	2009-09-28
Time limit:	4s
Source limit:	50000B
Languages:	All
Resource:	Kruzade 09

SPOJ Problem Set (classical)

4876. Save City

Problem code: KZCITY

In 1095 A.D, the crusade war broke.At 1156 the christians were at their peak made the jews to run to safe guard thier lives.
Their way of attack was different,they turned the cities into ashes.Think you are in the country "almoravids" where the lands weren't
too high from sea level,and also each city is like a square ,sliced into smaller identical squares by a regular grid of roads.
Every square contains one of three kind of terrain buliding,grassland or industry.You are a fireman and you are asked to safe-guard
the city from fire.You are provided only with explosives to extinguish the fire.
cruzade

If a city is blown which is either under fire or not,
a crater is formed and water from nearby sea enters in 2 minutes.The fire spreads only to horizontal or vertical city adjacent to it
in 2 minutes.When fire boke out you can hardly waste 2 seconds with your glider in the sky.If you drop a dynamite,
you can move to any city in random by 2 minutes.The loss by fire and dynamite are same they leave the city without any resources left.
Safe guard as many cities as possible.

Input:

the first line of input contains the the no.of cities along a row and no.of cities along a coloumn.The next line of input contains the row and coloumn of the city under fire(matrix representation).

Output

Output contains the cities atmost safeguarded.the next line contains the present situation of the land.a->turned into ashes,s->land is safe ,f->flooded by water.

Example

```
Input:

5 4 3
Output:
1s f f a a f a a a aa a a a a a a a a a a a a a a
```

Added by:	balaji
Date:	2009-09-28
Time limit:	5s
Source limit:	50000B
Languages:	All except: SCALA
Resource:	Kruzade 09 main

SPOJ Problem Set (tutorial)

4892. De minúsculas a mayúsculas

Problem code: VOCMINMA

Elabore un programa que dada una cadena de caracteres reemplace todas las vocales que se encuentren en minúscula por la misma vocal pero en mayúsculas.

Bono: La función realizada es manejada recursivamente.

Input

La entrada comienza con un número t indicando los casos de prueba. En cada caso hay una cadena de caracteres con 1<= n <=500000, siendo n la cantidad de caracteres. Dentro de esa cadena pueden haber espacios, signos de puntuación, entre otros.

Output

Para cada caso de prueba imprimir la nueva cadena que contiene todas sus vocales en mayúscula.

Example

```
Input:3Me dan mIEdo lOs murciElagosprogramaciOn de cOmPutadOResCamiLa! cuiDado en la pLAya!Output:ME dAn mIEdO lOs mUrcIElAgOsprOgrAmAcIOn dE cOmPUtAdOREsCAmILA! cUIDAdO En lA pLAyA!
```

Added by: Fabio Avellaneda
Date: 2009-10-01
Time limit: 1s
Source limit:50000B
Languages: C++

SPOJ Problem Set (tutorial)

4893. Traductor Simple

Problem code: SIMTRANS

Un traductor consta de una cadena que contiene una palabra en un idioma y seguida a ésta, la traducción de esa palabra a otro idioma. Si se tienen varias palabras en el traductor, su almacenamiento sería:

Hola Hi name nombre age edad

Elabore un programa que traduzca cadenas de un idioma a otro.

La calificación de este punto dependerá de cuantos casos de prueba sea capaz de resolver correctamente.

Input

La entrada consta de una cadena con tamano n (1<= n <= 500000) que contiene las palabras en los dos idiomas separadas por un espacio.

Seguida a esta cadena se presenta la cantidad c de cadenas a traducir, por lo que las siguientes c líneas serán cadenas de caracteres, todas en un mismo idioma.

Output

Para cada cadena se debe imprimir su respectiva traducción.

Example

```
Input:it ello Was era at en dark oscuro looked miraron they Ellos2it Was darkthey looked at itOutput:
Ello era oscuroEllos miraron en Ello
```

Added by: Fabio Avellaneda
Date: 2009-10-01
Time limit: 2s
Source limit:50000B
Languages: C++ LISP clisp

SPOJ Problem Set (classical)

4895. N-D Intervals

Problem code: NDINTERV

In this problem you will be given a list of **N (0<N<3200)** intervals, followed by a second list of **Q (0<Q<510000)** query intervals. Your job is to simply return the number of query intervals which intersect one or more of the **N** given intervals. However, the intervals in this problem can have dimension **D (0<D<5)**. A **d**-dimensional interval can be represented as **[(a1, a2, ..., ad), (b1, b2, .., bd)]**. A point **(v1, v2, .. vd)** is considered to be part of this interval if and only if **min(ai, bi) <= vi <= max(ai, bi)** for all **i** in **[1, d]**. Two intervals intersect if and only if they share at least one point. All interval coordinates are between **0** and **50**.

Input

A number of inputs (<5), each starting with **D, N** and **Q,**followed by 2***N** lines with **D** numbers on each line. Each pair of lines represent one interval. This is then followed by 2***Q** lines with **D** numbers on each line, with each pair representing a query interval

Output

For each set of inputs, return the number of query intervals which intersect at least one given interval.

Example

`Input:``4 2 31 1 1 15 5 5 52 2 2 26 6 6 67 7 7 78 8 8 89 9 9 910 10 10 101 2 3 44 2 2 1`**`Output:`**`1`

Added by: Chen Xiaohong
Date: 2009-10-02
Time limit: 7s
Source limit:50000B
Languages: All except: SCALA
Resource: original

SPOJ Problem Set (tutorial)

4897. Meowist Networking

Problem code: MEOWIST

Kat Mandu has been living in seclusion for several years, practicing martial arts and meditating. As a consequence, he missed the take-off of social networking. He was therefore surprised upon his return to society that countless many are standing with signs at every street corner, eager to be followed on Twitter; his grandmother runs one of the world’s top blogs; and chances are, even *your mom* has Facebook.

Reluctantly, he joined the craze, but he quickly discovered that many of these services are lacking in their friend sorting facilities: they will usually only allow listing friends in alphabetical order by name. Kat Mandu would prefer sorting his friends by age, with the oldest at the top of the list. If two friends have the same age, only then sort alphabetically. Help Kat Mandu by writing a program which implements this functionality.

Input

Each line of input will contain a name and an age, separated by a space. Names will be unique and consist of at most 10 uppercase letters. Ages will be between 1 and 100 inclusive.

Output

Print out the same names that were given in the input, but sorted according to Kat Mandu’s requirements.

Example

```
Input:
DUCHESS 26
MARIE 8
BERLIOZ 8
TOULOUSE 7
THOMAS 28

Output:
THOMAS
DUCHESS
BERLIOZ
MARIE
TOULOUSE
```

Added by:	Miorel Palii
Date:	2009-10-02
Time limit:	2s
Source limit:	4096B
Languages:	All except: ERL TECS
Resource:	University of Florida Local Contest - September 27, 2009

SPOJ Problem Set (tutorial)

4906. Pythagorean Triples

Problem code: PYTHTRIP

A Pythagorean triple (A, B, C) is defined as three positive integers that satisfy the Pythagorean Theorem: $A^2 + B^2 = C^2$. Given two positive integers A and B, your task is to verify whether they are the "legs" in a Pythagorean triple, i.e. if an integer C exists such that (A, B, C) is a Pythagorean triple.

Input

The first line will contain a single integer N (0 < N <= 10000). Each of the next N lines will contain two integers A and B (0 < A, B <= 100).

Output

For each test case, output a single line. If a valid C exists, output a line containing the word YES and the value of C, separated by a space. Otherwise, output the single word NO.

Example

Input:`42 24 34 55 12`**Output:**`NOYES 5NOYES 13`

Added by: Miorel Palii
Date: 2009-10-04
Time limit: 2s
Source limit:4096B
Languages: All except: ERL
Resource: University of Florida Local Contest - April 13, 2009

SPOJ Problem Set (tutorial)

4907. Most Common Letter

Problem code: MCL

Many word processors have a word count feature, which can tell you not only how many words are in a file, but also how many characters, lines, paragraphs, and pages. But who cares about that? All you really need to know is which of the 26 letters of the English alphabet (A - Z) you've used the most in your text. Write a program which implements this feature.

Input

The input will be several lines, each representing a different test case. No test case will exceed 1024 characters in length, and each will contain at least one alphabetic character.

Output

For each test case, output one line, containing the most common letter and the number of times it appears in the test case, separated by a space. Break ties in favor of the letter that occurs first alphabetically. Ignore non-alphabetic characters. Counting should be case-insensitive, and output should be uppercase.

Example

Input:
```
Hello World!
Never gonna give you up, never gonna let you down...
You just lost the game.
I'm going to sleep *yawn* ZZZzzz
```

Output:
```
L 3
N 7
T 3
Z 6
```

Added by:	Miorel Palii
Date:	2009-10-04
Time limit:	2s
Source limit:	4096B
Languages:	All except: SCALA
Resource:	University of Florida Local Contest - April 13, 2009

SPOJ Problem Set (tutorial)

4997. Test Básico de Listas

Problem code: LISTTEST

Input

Primero leer un número que indica cuantos casos ingresan.
Para cada caso leer la cantidad de datos que ingresan.
Para cada dato, leer el número a ingresar y si se debe insertar o anexar.

Output

Para cada caso se debe imprimir la lista resultante, cada valor separado por un espacio.

Example

`Input:``231 insertar2 InserTar3 anexar20 INSERTAR0 anexar`**`Output:`**`2 3 1 0 0`

Added by:	Fabio Avellaneda
Date:	2009-10-15
Time limit:	3s
Source limit:	50000B
Languages:	C++

SPOJ Problem Set (tutorial)

5009. Alphabet Arithmetic

Problem code: ALPHMATH

In his Introduction to Digital Arts and Sciences class, Dave frequently assigns homework. It usually consists of tedious exercises involving integer arithmetic. The students respond like true students: by complaining. So, to make the homeworks more exciting for his students, Dave has decided to disguise the exercises as alphabet arithmetic! This is exactly the same as integer arithmetic, except that instead of the usual digits 0 through 9, it uses letters A through J as the digits.

Help Dave generate the key to the homework!

Input

Each test case will be on one line of the form "NUMBER OP NUMBER". Each NUMBER is positive and less than 100000. OP is one of +, -, *, /. You will not have to divide by zero.

Output

For each test case, output the alphabet arithmetic answer on a single line.

Example

```
Input:
G * H
D - F
B + B
H / C

Output:
EC
-C
C
D
```

Added by: Miorel Palii
Date: 2009-10-16
Time limit: 1s
Source limit:4096B
Languages: All except: ERL
Resource: University of Florida Team Practice 2009

SPOJ Problem Set (tutorial)

5012. Spanish Conjugation

Problem code: SPANCONJ

Spanish Conjugation

As you want to go to Madrid, maybe now would be the right time to learn some Spanish grammar? Of course, as a programmer, one way to do so is to write a program that teaches you. In this task, you are required to write the part of a grammar trainer that checks whether you conjugated a verb correctly.

In Spanish, there are three forms of regular verbs: Those ending in -ar, -er and -ir. The verb forms are build according to the following table:

	-ar		**-er**		**-ir**	
	Singular	**Plural**	**Singular**	**Plural**	**Singular**	**Plural**
First person	-o	-amos	-o	-emos	-o	-imos
Second Person	-as	-Ais	-es	-Eis	-es	-Is
Third Person	-a	-an	-e	-en	-e	-en

As an example, the word comer (to eat) in second person singular would be the stem (com) and the ending (es): comes, you eat. Note that the pronoun tu (you) is implicit; unlike English, the person can be derived from the verb alone.

In order to avoid compatibility problems between computers of different character encodings, we substituted some letters. Throughout this problem, we will write A instead of á, E instead of é and I instead of í. All other non-ASCII characters have been replaced by their normalized version (e.g. n->n).

Input

The input contains of several test cases. Each test case consists of a verb conjugation.

Each conjugation consists of the infinitiv of a verb, a comma, the person (first, second, third person), the number (singular or plural), a colon and the conjugated verb form. Only regular verbs will occur.

Output

For each test case, print either "correct" or "incorrect, should be ___" (where ___ is replaced with the correct conjugation).

Example

Input:hablar, first person singular: hablobeber, second person plural: bebEiscomer, first person plural: comemasvivir, third person singular: vive**Output**:correctcorrectincorrect, should be comemoscorrect

---

Added by:	Jonas Wagner
Date:	2009-10-17
Time limit:	1s
Source limit:	50000B
Languages:	All except: ERL

SPOJ Problem Set (tutorial)

5194. Robbery

Problem code: ROB

k bandits robbed a bank. They took away **n** gold coins. Being a progressive group of robbers they decided to use the following procedure to divide the coins. First the most respected bandit takes **1** coin, then the second respected takes **2** coins, ..., the least respected takes **k** coins, then again the most respected takes **k+1** coins, ans so on, until one of the bandits takes the remaining coins. Calculate how much gold each of the bandits gets.

Input

The first line of the input contains number **t** - the amount of tests. Then **t** test descriptions follow. Each test consists of two integers **n** and **k** - the amount of coins and bandits respectively.

Constraints

1 <= **t** <= 500
> 1 <= **n<b> <= 10^9**
> 2 <= k<b> <= 100

Output

For each test print the amounts of coins each bandit gets separated by spaces.

Example

```
Input:
3
10 2
11 3
12 4

Output:
4 6
5 3 3
3 2 3 4
```

Added by:	**Spooky**
Date:	**2009-11-03**
Time limit:	**1s**
Source limit:	**50000B**
Languages:	**All**
Resource:	**Advancement Autumn 2009, http://sevolymp.uuuq.com/**

SPOJ Problem Set (tutorial)

5195. Angry Knights

Problem code: ANGRYKN

Some angry knights want to settle on the checkmate board of **n x m** size. The angry knights are much like the ordinary ones, but each can move at any time. Moreover they don't like each other and won't allow any other knight on their territory. Luckily someone removed some cells from the board, so now more knights can settle on the board without bothering each other. Count the maximal number of angry knight that can live on the board simultaneously.

Input

The first line of the input contains number **t** - the amount of tests. Then **t** test descriptions follow. Each test starts with two numbers **n** and **m** - the dimensions of the board. Then **n** lines follow each consisting of **m** characters. Character 'x' means that the corresponding cell is removed, character '.' that it is present.

Constraints

1 <= **t** <= 100
2 <= **n**, **m** <= 100

Output

For each test print the maximal number of angry knights that can settle on such board.

Example

```
Input:
2
2 3
...
...
3 3
...
xxx
...

Output:
4
2
```

Added by:	Spooky
Date:	2009-11-03
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	Advancement Autumn 2009, http://sevolymp.uuuq.com/, author: Alexey Shchepin

SPOJ Problem Set (tutorial)

5200. Lsu Football

Problem code: BRHFBALL

Butch unfortunately missed the most recent LSU football game, but he was luckily able to get the score S (0 <= S <= 30) from a friend. So he got to thinking, how many possible ways could LSU have scored this?
Remember that the ways to score are like so:
2 - Safety
3 - Field Goal
6 - Touchdown with missed extra point or failed conversion (only include one 6-point in your calculations; see note below)
7 - Touchdown with the extra point
8 - Touchdown with a 2 point conversion
Butch would figure out how many ways himself, but he’s busy scouring the web for a replay, so he wants you to help.
Note: The order is important. For example, if the input is 5, there would be **2** ways: LSU could score a safety and then a field goal, or it could score a field goal and then a safety.

Note about the 6 point-ers: For example 8 points in total, the number of ways to score would be:

2-2-2-2

2-3-3

3-2-3

3-3-2

2-6

6-2

8

See that there is only one set of "6-2" and "2-6"; in other words, we don’t say "they scored from a safety and they scored from a touchdown with a failed extra point" and "they scored from a safety and they scored from a touchdown with a failed conversion", etc... From Neal, "You should not consider scoring the touchdown and missing the extra point, and scoring the touchdown and failing the conversion as two separate ways to score."

Another note: Values may not be precalculated and stored in an array. ***Any solution that does this will be disqualified and receive 0 points.***

Extra Challenge:The last test case will have S <= 10000.

Find the number of ways that a score S can be made in a football game, modulo 10000.

Input

Line 1: A single integer, S

Output

Line 1: A single integer, how many ways they could score

Example

```
Input:
8Output:
7
```

Added by: Damon Doucet
Date: 2009-11-04
Time limit: 5s
Source limit:50000B
Languages: All

SPOJ Problem Set (classical)

5228. Old problem

Problem code: OLDP

Let G is the convex polygon with area S and perimeter L. We need to know volume of set of points which distance from G is not greater than R.

Input

Number of test cases in first line and three integers S, L and R for each test case. All integers in input are nonnegative and less than 100.

Output

Volume for each test case with 10^{-2} precision.

Example

```
Input:
1
48 57 1

Output:
189.724
```

Added by: Ruslan Sennov
Date: 2009-11-07
Time limit: 1s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

5241. Alchemy

Problem code: ALCH

Many computer games implement alchemy skill. It allows the player to create different elixirs using various ingredients. Usually players have to find out recipes for the elixirs they need. Usually they do it by trying to mix some ingredients. Given that there are **n** different ingredients and any elixir can be made by mixing three or more different ingredients, can you count the maximal number of various elixirs that can be made using alchemy skill.

Input

The first line of the input contains number **t** - the amount of tests. Then **t** test descriptions follow. Each test consist of a single integer **n**.

Constraints

1 <= **t** <= 10000
1 <= **n** <= 10^9

Output

For each test print the maximal number of different elixirs that can be made modulo **1000000007**.

Example

```
Input:
4
3
4
100
100000

Output:
1
5
976366234
607673554
```

Added by: Spooky
Date: 2009-11-07
Time limit: 1s
Source limit:50000B
Languages: All
Resource: Advancement Autumn 2009, http://sevolymp.uuuq.com/

SPOJ Problem Set (tutorial)

5302. Surprising Strings

Problem code: MCPC06C

The D-pairs of a string of letters are the ordered pairs of letters that are distance D from each other. A string is D unique if all of its D-pairs are different. A string is surprising if it is D-unique for every possible distance D.

Consider the string ZGBG. Its 0-pairs are ZG, GB, and BG. Since these three pairs are all different, ZGBG is 0 unique. Similarly, the 1-pairs of ZGBG are ZB and GG, and since these two pairs are different, ZGBG is 1-unique. Finally, the only 2-pair of ZGBG is ZG, so ZGBG is 2-unique. Thus ZGBG is surprising. (Note that the fact that ZG is both a 0-pair and a 2-pair of ZGBG is irrelevant, because 0 and 2 are different distances.)

Input

The input consists of one or more nonempty strings of at most 79 uppercase letters, each string on a line by itself, followed by a line containing only an asterisk that signals the end of the input.

Output

For each string of letters, output whether or not it is surprising using the *exact* output format shown below.

Example

Input: ZGBGXEEAABAABAAABBBCBABCC*

Output:
ZGBG is surprising.X is surprising.EE is surprising.AAB is surprising.AABA is surprising.AABB is NOT surprising.BCBABCC is NOT surprising.

Added by: Tamer
Date: 2009-11-15
Time limit: 3s
Source limit:50000B
Languages: All
Resource: ACM Mid-Central Programming Contest 2006

SPOJ Problem Set (classical)

5306. Linear Pachinko

Problem code: MCPC06B

This problem is inspired by Pachinko, a popular game in Japan. A traditional Pachinko machine is a cross between a vertical pinball machine and a slot machine. The player launches small steel balls to the top of the machine using a plunger as in pinball. A ball drops through a maze of pins that deflect the ball, and eventually the ball either exits at a hole in the bottom and is lost, or lands in one of many gates scattered throughout the machine which reward the player with more balls in varying amounts. Players who collect enough balls can trade them in for prizes.

For the purposes of this problem, a linear Pachinko machine is a sequence of one or more of the following: holes (."), floor tiles (_"), walls (|"), and mountains (/\"). A wall or mountain will never be adjacent to another wall or mountain. To play the game, a ball is dropped at random over some character within a machine. A ball dropped into a hole falls through. A ball dropped onto a floor tile stops immediately. A ball dropped onto the left side of a mountain rolls to the left across any number of consecutive floor tiles until it falls into a hole, falls off the left end of the machine, or stops by hitting a wall or mountain. A ball dropped onto the right side of a mountain behaves similarly. A ball dropped onto a wall behaves as if it were dropped onto the left or right side of a mountain, with a 50% chance for each. If a ball is dropped at random over the machine, with all starting positions being equally likely, what is the probability that the ball will fall either through a hole or off an end? For example, consider the following machine, where the numbers just indicate character positions and are not part of the machine itself:

```
123456789
/\.|__/\.
```

The probabilities that a ball will fall through a hole or off the end of the machine are as follows, by position: 1 = 100%, 2 = 100%, 3 = 100%, 4 = 50%, 5 = 0%, 6 = 0%, 7 = 0%, 8 = 100%, 9 = 100%. The combined probability for the whole machine is just the average, which is approximately 61.111%.

Input

The input consists of one or more linear Pachinko machines, each 1âEuro"79 characters long and on a line by itself, followed by a line containing only "#" that signals the end of the input.

Output

For each machine, compute as accurately as possible the probability that a ball will fall through a hole or off the end when dropped at random, then output a single line containing that percentage truncated to an integer by dropping any fractional part.

Example

Input:
```
/\.|__/\.
_._/\_|.__/\./\_
...
___
./\.
_/\_
_|.|_|.|_|.|_
____|_____
#
```

Output:
```
61
53
100
0
100
50
53
10
```

Added by: Tamer
Date: 2009-11-15
Time limit: 5s
Source limit:50000B
Languages: All
Resource: Mid Central Regional Contest 2006

SPOJ Problem Set (classical)

5313. Root of the Problem

Problem code: MCPC06G

Given positive integers B and N, find an integer A such that A^N is as close as possible to B. (The result A is an approximation to the Nth root of B.) Note that A^N may be less than, equal to, or greater than B.

Input

The input consists of one or more pairs of values for B and N. Each pair appears on a single line, delimited by a single space. A line specifying the value zero for both B and N marks the end of the input. The value of B will be in the range 1 to 1,000,000 (inclusive), and the value of N will be in the range 1 to 9 (inclusive).

Output

For each pair B and N in the input, output A as defined above on a line by itself.

Example

Input:`4 35 327 3750 51000 52000 53000 51000000 50 0`

Output:
`123444516`

Added by: Tamer
Date: 2009-11-16
Time limit: 1s
Source limit:50000B
Languages: All
Resource: ACM Mid-Central Regional Programming Contest 2006

SPOJ Problem Set (tutorial)

5965. The Element of Surprise

Problem code: SURPRISE

General Tontus holds the record for most consecutive losses in military history. The emperor is not pleased that his general holds this unique distinction and has threatened to fire him in the event of another loss. The general is convinced that his losing streak is not due to poor strategy on his part. Instead, he blames his messengers for allowing important orders to be intercepted by the enemy, thus giving away the element of surprise. To alleviate this problem, the general has devised a cunning plan. All sensitive messages will have the order of their characters reversed, so that even if they fall in enemy hands, they will be incomprehensible! Because this scheme is too complex to be applied by hand, the general needs some way to automate it. Help save the general's job by writing a program to apply this encryption algorithm.

Input

The first line will contain a single integer N (0 < N < 100). Each of the next N lines will contain a string of at most 1024 characters which must be encrypted.

Output

For each test case, output a single line containing the encrpyted string. Beware, spaces are significant.

Example

Input:

```
4
 Hello World!
I know 10 digits of pi after the decimal: 3.1415926535
You just lost the game.
ATTACK AT DAWN
```

Output:

```
!dlroW olleH
5356295141.3 :lamiced eht retfa ip fo stigid 01 wonk I
.emag eht tsol tsuj uoY
NWAD TA KCATTA
```

Added by: Miorel Palii
Date: 2010-01-24
Time limit: 1s
Source limit:4096B
Languages: All except: TECS
Resource: own problem statement; used in practice round of a few contests

SPOJ Problem Set (tutorial)

5966. Generalized Chess

Problem code: GENCHESS

The emperor's younger brother, Minimus, is an excellent chess player who has never lost a game. Let us ignore for the moment the fact that anyone who might dare defeat him would likely suffer a horrible death. Having mastered the usual game, which is played on a board of size 8, Minimus wants to generalize chess so that it can be played on a square board of arbitrary size. Unfortunately, Minimus skipped class so he never learned his multiplication tables, so he'll give you a number N, and it will be up to you to tell him how many squares a chessboard of that size would have. Don't worry, Minimus can't count higher than a million, so that's the highest number you need to be able to handle.

Input

There will be several test cases, each consisting of a single positive integer on a separate line, representing a possible value of N. A value of zero indicates the end of input and should not be processed.

Output

For each test case, output a single line containing the number of squares on a chessboard of size N.

Example

```
Input:
8
1
42
999999
0

Output:
64
1
1764
999998000001
```

Added by: Miorel Palii
Date: 2010-01-24
Time limit: 1s
Source limit:4096B
Languages: All except: TECS
Resource: own problem statement; used in practice round of a few contests

SPOJ Problem Set (tutorial)

6291. Problem

Problem code: PROBLEM

This problem has no statement

Input

The format of the input data in not known.

Constraints

The input file is of the reasonable size.

Output

Output the right answer for each test.

Example

```
Input:
4
188
432
100
765

Output:
4
0
2
1
```

Added by: Spooky
Date: 2010-03-09
Time limit: 1s
Source limit:50000B
Languages: All except: TECS
Resource: Advancement Spring 2010, http://sevolymp.uuuq.com/

SPOJ Problem Set (tutorial)

6475. Da.o cho+i băng xe bus

Problem code: KMBUS

Môt tuye^’n đu+o+‘ng o+? thanh phô có các be^’n xe bus o+? tu+‘ng km tuye^’n đu+o+‘ng. Môi lân qua be^’n, xe đe^‘u đô đe^? đón khách. Môi be^’n đe^‘u có đie^?m xuât phát. Môt xe chi? cha.y không quá B km ke^? tu+‘ đie^?m xuât phát cu?a nó. Hanh khách khi đi xe se~ pha?i tra? tie^‘n cho đô dai đoa.n đu+o+‘ng ma ho. ngôi tren xe. Cu+o+’c phí cân tra? đe^? đi đoa.n đu+o+‘ng đô dai i la Ci(i=1,2..B). Môt du khách xuât phát tu+‘ 1 be^’n nao đó muôn đi da.o L km theo tuye^’n nói tren. Ho?i ông ta pha?i len xuông xe nhu+ the^’ nao đe^? tông sô tie^‘n pha?i tra? la nho? nhât có the^?.

Du+~ lie^.u vao: Dong đâu ghi 2 sô nguyen du+o+ng B, L.Dong thu+’ i trong sô B dong tie^’p theo ghi 1 sô nguyen du+o+ng Ci (1 <= i <= B).**Ke^’t qu?a**Môt dong duy nhât la sô tie^‘n nho? nhât pha?i tra?**Gio+’i ha.n** 0 <= B <= 1000 <= L <= 100000 <= Ci <= 100 **Ví du. Du+~ lie^.u:**5 73 46922**Ke^’t qu?a**14

Added by: Nguye^~n Tie^’n Hoang
Date: 2010-04-08
Time limit: 1s
Source limit:50000B
Languages: All except: TECS
Resource: Bai co+ ba?n.

SPOJ Problem Set (tutorial)

6501. Median

Problem code: XMEDIAN

Given an array x of n elements find the medians of its first k elements for each k from 1 to n inclusive. The median of an array is the middle element of that array when it is sorted. If there are an even number of elements in the array, we choose the first of the middle two elements to be the median.

Input

The first line of input contains number n (1 <= n <= 200000) - the amount of elements in the array. The next n lines contain the elements xi (1 <= xi <= 1000000).

Output

Output n integers - the medians of the first k elements of the array for each k from 1 to n inclusive.

Example

```
Input:
5
1
2
3
4
5

Output:
1
1
2
2
3
```

Added by: Spooky
Date: 2010-04-13
Time limit: 2s
Source limit:50000B
Languages: All except: TECS

SPOJ Problem Set (tutorial)

6521. SuperPower

Problem code: SUPERPW

Example

Added by: .::: Pratik :::.
Date: 2010-04-16
Time limit: 10s
Source limit:50000B
Languages: All except: TECS

SPOJ Problem Set (tutorial)

6523. Digital Root Counter

Problem code: DIGRT

The digital root of a number x is calculated by summing up all digits x, then adding all digits of the sum and so on, until we are left with only a single digit.

For example, if x = 987654, then its digital root d(x) = 9 + 8 + 7 + 6 + 5 + 4 = 39.
Now we sum up digits of 39. d(39) = 3 + 9 = 12.
Now we sum up digits of 12. d(12) = 1 + 2 = 3.
Thus, d(987654) = 3.

For example, if x = 987654, then its digital root d(x) = 9 + 8 + 7 + 6 + 5 + 4 = 39.

Now we sum up digits of 39. d(39) = 3 + 9 = 12.

Now we sum up digits of 12. d(12) = 1 + 2 = 3.

Thus, d(987654) = 3.

Input

The first line gives the number of test cases T (T <= 100).

Each of the next T test cases gives the value of N (N <= 50,000).

Output

For each test case, print 10 integers, each separated by a space, indicating number of digital roots in the range of 1 to N that are equal to i (1 <= i <= 9). Seperate each test case with a new line.

Example

`Input:21240`**`Output:`**`2 2 2 1 1 1 1 1 15 5 5 5 4 4 4 4 4`

Explanation:

For the first case, there are 2 numbers in the range of 1 to 12 that have digital root equal to 1 (1 and 10), 2 numbers that have digital root equal to 2 (2 and 11) and so on.

Added by: .::: Pratik :::.
Date: 2010-04-16
Time limit: 10s
Source limit:50000B
Languages: All except: TECS

SPOJ Problem Set (tutorial)

6524. Find the group

Problem code: FGROUP

You are given 2 arrays of size N and M.You have to answer Q queries where each query consists of a number.
You have to find out in which of the two arrays is the given number in.

You are given 2 arrays of size N and M.You have to answer Q queries where each query consists of a number.

You have to find out in which of the two arrays is the given number in.

Input

The first line gives the number of test cases T (T <= 10).

Then T test case follow.

The first line of each test case gives the value of N.The second line contains N space separated integers.

The third line of each test case gives the value of M.The fourth line contains M space separated integers.

Next line gives the value of Q, number of queries to be answered.

Q lines follow containing a number each on one line.

Output

For each test case, in response to the Q queries for that test-case print Q lines.

If the number is present in both arrays, print "both" (without quotes).

Else print the array in which the number is in (1 or 2).

If the number is not present in both arrays,print -1.

Print a new line after every test case.

Example

Input:

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

Output:

```
1
both
2
-1

both
both
both
```

Constraints:

```
N,M <= 20000
Q <= 30000
All numbers in the input will be less than 50000.
```

Explanation:

For the first case, 5 is present in array 1 only, 8 is present in both arrays, 6 is present in array 2 only while 6 is not present in either array.

Added by:	.::: Pratik :::.
Date:	2010-04-16
Time limit:	3s-15s
Source limit:	50000B
Languages:	All except: TECS

SPOJ Problem Set (tutorial)

6528. Circle Counting

Problem code: CIRCP

In this problem, you will be given N circles, and M points. You are required to find out that inside how many circles does each of the M points lie. None of the M points will lie on any of the N circle boundaries (they will either lie inside a circle or outside it, but not on the circle).

Input

The 1st line gives the number of test cases T.
Each of the next T test cases has a format as explained below.
The 1st line of each test case contains 2 integers N and M.
Each of next N lines contains 3 integers cx, cy, r. The circle is located at center (cx, cy) and has a radius r.
Each of the following M lines contain 2 integers (x, y), which indicate that the location of the point.

The 1st line gives the number of test cases T.

Each of the next T test cases has a format as explained below.

The 1st line of each test case contains 2 integers N and M.

Each of next N lines contains 3 integers cx, cy, r. The circle is located at center (cx, cy) and has a radius r.

Each of the following M lines contain 2 integers (x, y), which indicate that the location of the point.

Output

For each of test case, print M lines, each correspoding to the no of circles inside which the corresponding point lies.

Separate each test case with a blank line.

Example

Input:
```
1
2 3
3 3 2
1 1 2
1 1
2 2
3 3
```

Output:
```
1
2
1
```

Sample diagram

Constraints
```
T <= 100
0 <= N <= 500
0 <= M <= 500
-1000 <= cx <= 1000
-1000 <= cy <= 1000
-1000 <= x <= 1000
-1000 <= y <= 1000
1 <= r <= 1000
```

Added by: .::: Pratik :::.
Date: 2010-04-17
Time limit: 10s
Source limit:50000B
Languages: All except: TECS

SPOJ Problem Set (challenge)

6555. F - Rompecabezas de Puzzle

Problem code: BOPUZZLE

Problema F - Rompecabezas de Puzzle

El rompecabezas de puzzle consiste en una base de 6 posiciones en los vértices de un hexágono regular y otra posición en el centro, conectado como se muestra en la figura de abajo. Hay seis fichas marcadas A, B, C, D, E y F. Un solo movimiento del puzzle es mover una ficha a una posición adyacente vacía (a lo largo de una conexión permitida - los segmentos de línea en el diagrama a continuación). La idea del rompecabezas es empezar con un arreglo inicial de fichas con el centro vacío y, por una secuencia de movimientos, llegar a la configuración de la figura de abajo.

Una posición inicial para el rompecabezas viene dada por una permutación de las letras A a la F. La primera letra comienza en A en la figura, el próximo es B y así sucesivamente. Una secuencia de movimientos se especifica mediante la lista de las etiquetas de las fichas que va a mover en el orden en que han de ser movido.

Por ejemplo, para resolver el rompecabezas FACDBE, utiliza los movimientos BEFAB.

Nota: No todas las permutaciones de partida puede ser resuelto.

Escriba un programa que, dada una permutación inicial, encuentre la menor secuencia de movimientos para resolver el rompecabezas o determine que no existe una solución.

Entrada

La primera línea de entrada contiene un entero P, (1 <= P <= 1000), que es el número de conjuntos de datos. Cada conjunto de datos es una línea que contiene el número de conjunto de datos, seguido de un espacio, seguido por una permutación de las letras A a F representando la posición inicial del rompecabezas.

Salida

Para cada conjunto de datos hay una sola línea de salida. Si no hay solución, la línea contiene un decimal
entero representando al número de conjunto de datos seguido de un espacio único, seguido de la cadena NO SOLUTION. Si hay una solución, la línea contiene un entero decimal representando el número de conjuntos de datos seguido por un espacio, seguido por el número de movimientos en la solución, seguido de un espacio, seguido por la solución como una cadena de letras de la A a la F. Si el número de movimientos es cero (0), tu aún deberías mostrar el espacio después del 0, a pesar de que no existe una cadena de letras.

Ejemplo de Entrada	Ejemplo de Salida
12 1 FACDBE 2 ABCDEF 3 ADCEFB 4 ADCEBF 5 FEDCBA 6 FEDCAB 7 ECBFAD 8 ECBFDA 9 DCEBFA 10 DCEBAF 11 CBEADF 12 BDEAFC	1 5 BEFAB 2 0 3 19 DABFECABFEDBACDEFAB 4 NO SOLUTION 5 29 BCDEBCAFBCAFBCEDFAECBAFDCBAFE 6 NO SOLUTION 7 19 CBFACBFACDEFACDEFAB 8 NO SOLUTION 9 13 CDAFBEDCBEDCB 10 NO SOLUTION 11 21 DAEBDAEBDCFEBDCABEFAB 12 16 FAEDBCAFBCAFEDCB

Added by:	Alvaro
Date:	2010-04-21
Time limit:	2s
Source limit:	50000B
Languages:	C++ 4.0.0-8 C++ 4.3.2 JAVA

SPOJ Problem Set (tutorial)

6651. Snapper Chain

Problem code: GCJ2010

Problem

The *Snapper* is a clever little device that, on one side, plugs its input plug into an output socket, and, on the other side, exposes an output socket for plugging in a light or other device.

When a *Snapper* is in the ON state and is receiving power from its input plug, then the device connected to its output socket is receiving power as well. When you snap your fingers -- making a clicking sound -- any *Snapper* receiving power at the time of the snap toggles between the ON and OFF states.

In hopes of destroying the universe by means of a singularity, I have purchased **N** *Snapper* devices and chained them together by plugging the first one into a power socket, the second one into the first one, and so on. The light is plugged into the **N**th *Snapper*.

Initially, all the *Snapper*s are in the OFF state, so only the first one is receiving power from the socket, and the light is off. I snap my fingers once, which toggles the first *Snapper* into the ON state and gives power to the second one. I snap my fingers again, which toggles both *Snapper*s and then promptly cuts power off from the second one, leaving it in the ON state, but with no power. I snap my fingers the third time, which toggles the first *Snapper* again and gives power to the second one. Now both *Snapper*s are in the ON state, and if my light is plugged into the second *Snapper* it will be *on*.

I keep doing this for hours. Will the light be *on* or *off* after I have snapped my fingers **K** times? The light is *on* if and only if it's receiving power from the *Snapper* it's plugged into.

Input

The first line of the input gives the number of test cases, **T**. **T** lines follow. Each one contains two integers, **N** and **K**.

Output

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is either "ON" or "OFF", indicating the state of the light bulb.

Limits

1 <= **T** <= 10,000.

Small dataset

1 <= **N** <= 10;
0 <= **K** <= 100;

Large dataset

1 <= **N** <= 30;
0 <= **K** <= 10^8;

Sample

4

1 0

1 1

4 0

4 47

Output:

Case #1: OFF

Case #2: ON

Case #3: OFF

Case #4: ON

Added by:	Kumar Anurag
Date:	2010-05-14
Time limit:	1s
Source limit:	500B
Languages:	All except: TECS
Resource:	Google Code Jam 2010 Qualifying round

SPOJ Problem Set (tutorial)

6653. Lexicographic Order 1

Problem code: LEXI1

An ordering for the Cartesian product x of any two sets A and B with order relations <A and <B, respectively, such that if (a1, b1) and (a2, b2) both belong to AxB, then (a1, b1) < (a2, b2) iff either

- a1 <A a2, or
- a1 = a2 and b1 <B b2.

The lexicographic order can be readily extended to cartesian products of arbitrary length by recursively applying this definition, i.e., by observing that AxBxC = Ax(BxC).

When applied to permutations, lexicographic order is increasing numerical order. For example, the permutations of {1,2,3} in lexicographic order are 123, 132, 213, 231, 312, and 321.

You will be given a permutation of n first natural numbers. You need to find k-th lexicographically next permutaion. Also we will consider that the lexicographically last permutaion is followed by the first one in the ordering.

Input

The first line is number t - the amount of test cases. Each test case starts with numbers n and k. Then n natural numbers follow which are the elements of the given permutation.

Constraints

1 <= t <= 5
1 <= n <= 50000
0 <= k <= 100

Output

For each test case output the k-th lexicographically next permutation after the given one.

Example

Input:
```
3
3 3
1 2 3
3 2
2 1 3
3 8
2 3 1
```

Output:
```
2 3 1
3 1 2
3 2 1
```

Added by:	Spooky
Date:	2010-05-15
Time limit:	2s
Source limit:	50000B
Languages:	All except: TECS

SPOJ Problem Set (tutorial)

6654. Lexicographic Order 2

Problem code: LEXI2

An ordering for the Cartesian product x of any two sets A and B with order relations <A and <B, respectively, such that if (a1, b1) and (a2, b2) both belong to AxB, then (a1, b1) < (a2, b2) iff either

- a1 <A a2, or
- a1 = a2 and b1 <B b2.

The lexicographic order can be readily extended to cartesian products of arbitrary length by recursively applying this definition, i.e., by observing that AxBxC = Ax(BxC).

When applied to permutations, lexicographic order is increasing numerical order. For example, the permutations of {1,2,3} in lexicographic order are 123, 132, 213, 231, 312, and 321.

You will be given a permutation of n first natural numbers. You need to find k-th lexicographically next permutaion. Also we will consider that the lexicographically last permutaion is followed by the first one in the ordering.

Input

The first line is number t - the amount of test cases. Each test case starts with numbers n and k. Then n natural numbers follow which are the elements of the given permutation.

Constraints

1 <= t <= 100
1 <= n <= 20
0 <= k < n!

Output

For each test case output the k-th lexicographically next permutation after the given one.

Example

```
Input:
3
3 3
1 2 3
3 2
2 1 3
3 5
2 3 1
```

Output:
```
2 3 1
3 1 2
2 1 3
```

Added by:	Spooky
Date:	2010-05-15
Time limit:	1s
Source limit:	50000B
Languages:	All except: TECS

SPOJ Problem Set (tutorial)

6655. Lexicographic Order 3

Problem code: LEXI3

An ordering for the Cartesian product x of any two sets A and B with order relations <A and <B, respectively, such that if (a1, b1) and (a2, b2) both belong to AxB, then (a1, b1) < (a2, b2) iff either

- a1 <A a2, or
- a1 = a2 and b1 <B b2.

The lexicographic order can be readily extended to cartesian products of arbitrary length by recursively applying this definition, i.e., by observing that AxBxC = Ax(BxC).

When applied to permutations, lexicographic order is increasing numerical order. For example, the permutations of {1,2,3} in lexicographic order are 123, 132, 213, 231, 312, and 321.

You will be given a permutation of n first natural numbers. You need to find k-th lexicographically next permutaion. Also we will consider that the lexicographically last permutaion is followed by the first one in the ordering.

Input

The first line is number t - the amount of test cases. Each test case starts with numbers n and k. Then n natural numbers follow which are the elements of the given permutation.

Constraints

1 <= t <= 100
1 <= n <= 250
0 <= k < n!

Output

For each test case output the k-th lexicographically next permutation after the given one.

Example

```
Input:
3
3 3
1 2 3
3 2
2 1 3
3 8
2 3 1
```

Output:
```
2 3 1
3 1 2
3 2 1
```

Added by:	Spooky
Date:	2010-05-15
Time limit:	2s
Source limit:	50000B
Languages:	All except: TECS

SPOJ Problem Set (tutorial)

6656. Lexicographic Order 4

Problem code: LEXI4

An ordering for the Cartesian product x of any two sets A and B with order relations <A and <B, respectively, such that if (a1, b1) and (a2, b2) both belong to AxB, then (a1, b1) < (a2, b2) iff either

- a1 <A a2, or
- a1 = a2 and b1 <B b2.

The lexicographic order can be readily extended to cartesian products of arbitrary length by recursively applying this definition, i.e., by observing that AxBxC = Ax(BxC).

When applied to subsets, two subsets are ordered by their smallest elements. For example, the subsets of {1,2,3} in lexicographic order are {}, {1}, {1,2}, {1,2,3}, {1,3}, {2}, {2,3}, {3}.

You will be given a subset of a set of first n natural numbers. You need to find k-th lexicographically next subset. Also we will consider that lexicographically last subset is followed by the first one in the ordering.

Input

The first line is number t - the amount of test cases. Each test case starts with numbers n and k. The next line describes the given subset. The description starts with number q - the amount of elements in the subset, followed by q natural numbers - the elements of the subset.

Constraints

1 <= t <= 5
1 <= n <= 50000
0 <= k <= 10000
0 <= q <= n

Output

For each test case output the k-th lexicographically next subset after the given one. If the result is an empty set then print "empty".

Example

```
Input:
3
3 1
1 3
3 3
2 1 3
5 5
0
```

Output:

```
empty
3
1 2 3 4 5
```

Added by:	Spooky
Date:	2010-05-16
Time limit:	1s
Source limit:	50000B
Languages:	All except: TECS

SPOJ Problem Set (tutorial)

6677. Rope Intranet

Problem code: GCJ2K10

Problem

A company is located in two very tall buildings. The company intranet connecting the buildings consists of many wires, each connecting a window on the first building to a window on the second building.

You are looking at those buildings from the side, so that one of the buildings is to the left and one is to the right. The windows on the left building are seen as points on its right wall, and the windows on the right building are seen as points on its left wall. Wires are straight segments connecting a window on the left building to a window on the right building.

You've noticed that no two wires share an endpoint (in other words, there's at most one wire going out of each window). However, from your viewpoint, some of the wires intersect midway. You've also noticed that exactly two wires meet at each intersection point.

On the above picture, the intersection points are the black circles, while the windows are the white circles.

How many intersection points do you see?

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow. Each case begins with a line containing an integer **N**, denoting the number of wires you see.

The next **N** lines each describe one wire with two integers $\mathbf{A_i}$ and $\mathbf{B_i}$. These describe the windows that this wire connects: $\mathbf{A_i}$ is the height of the window on the left building, and $\mathbf{B_i}$ is the height of the window on the right building.

Output

For each test case, output one line containing "Case #x: y", where x is the case number (starting from 1) and y is the number of intersection points you see.

Limits

1 <= **T** <= 15.
1 <= $\mathbf{A_i}$ <= 10^4.
1 <= $\mathbf{B_i}$ <= 10^4.
Within each test case, all $\mathbf{A_i}$ are different.
Within each test case, all $\mathbf{B_i}$ are different.

No three wires intersect at the same point.

Small dataset

1 <= **N** <= 2.

Large dataset

1 <= **N** <= 1000.

Inputs:

2
3
1 10
5 5
7 7
2
1 1
2 2

output:

```
Case #1: 2
Case #2: 0
```

Added by:	Kumar Anurag
Date:	2010-05-23
Time limit:	1s
Source limit:	5000B
Languages:	All except: TECS
Resource:	Google Code Jam 2010 Round 1

SPOJ Problem Set (challenge)

6696. F - The Sweeper in Cleanland

Problem code: BOSWEEP

PROBLEM F

THE SWEEPER IN CLEANLAND

Cleanland is a city well-known for the civility of its inhabitants. There is no crime, the streets are clean, drivers respect speed limits. In spite of its cleanliness, every day, during the early hours of the morning, a machine is used to clean the curb sides of each street. The city is not too large, so every street is a two-way street, and there is only one machine available to do the job of sweeping the streets. Both sides of each street must be swept. The operator of the sweeping machine is a inhabitant of the city, therefore the curb is traversed in the direction the traffic flows, so each street must be traversed in both directions, cleaning one side at a time.

Your job is to provide the operator with an optimal route, so that no street is traversed more than twice, once in each direction. Of course, one may drive from any point in the city to any other point in that same town. For simplicity, we assume that every street is one block in length, connecting two distinct *corners*. The figure below gives a small example one such town, where is an optimal route.

Input

The input contains several test cases, each case consists of two or more lines. The first line contains two integers, n and m, separated by one space. Integer n, (2<=n<=100), is the number of corners, which are identified by the integers 0, 1, . . . , n-1. Integer m, (0 < m <= n(n-1)/2), is the number of streets. The next m lines describe the m streets: every line contains two integers v and w, separated by one space, corresponding to the corners connected by the street. Note that no two corners are connected by two or more streets. Note also that the ends of each street are distinct. The end of the input is indicated by a line containing precisely two zeros, separated by one space.

Output

For each test case, your program should output one line. The line contains case c: followed by one space, where c is the case number, without leading zeros (the first case is case number 1). Then, separated by precisely one space, an optimal route, described by its corners, in the order they appear in the route. You should not repeat the initial corner as the last corner. All these integers should be printed in the same line, separated by exactly one space.

Sample Input

6 7

0 1

0 5

1 2

1 5

2 3

2 4

3 4

2 1

0 1

0 0

Sample Output

case 1: 2 3 4 2 4 3 2 1 0 5 1 5 0 1

case 2: 0 1

Notice:

Imagen3

Figure 1: An optimal route: 2 3 4 2 4 3 2 1 0 5 1 5 0 1

Added by: Alvaro
Date: 2010-05-25
Time limit: 1s
Source limit:50000B
Languages: C C++ 4.0.0-8 C++ 4.3.2 JAVA

SPOJ Problem Set (classical)

6734. Crossing the bridge

Problem code: CRUCE

n people wish to cross a bridge at night. A group of at most two people may cross at any time, and each group must have a flashlight. Only one flashlight is available among the *n* people, so some sort of shuttle arrangement must be arranged in order to return the flashlight so that more people may cross.

Each person has a different crossing speed; the speed of a group is determined by the speed of the slower member. Your job is to determine a strategy that gets all *n* people across the bridge in the minimum time.

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

The line of input contains *n*, followed by *n* lines giving the crossing times for each of the people. There are not more than 1000 people and nobody takes more than 100 seconds to cross the bridge.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

The line of output must contain the total number of seconds required for all *n* people to cross the bridge.

Example

Input:

1
4
1
2
5
10

Output

17

Added by:	Coach UTN FRSF
Date:	2010-06-03
Time limit:	1s-3s
Source limit:	50000B
Languages:	All except: TECS
Resource:	http://uva.onlinejudge.org/external/100/10037.html

SPOJ Problem Set (classical)

7129. Happy Telephones

Problem code: TTREAT

In the land of Eden, all phone conversations are happy ones. People com-plaining on the phone are immediately put in jail. To enforce this law, thepolice taps all phone conversations.The police wants to hire the approriate number of operators to listen to allconversations in a given period of time. Unfortunately, each of their operatorscan listen to one conversation only before needing a really long break to restfrom the eort.As a contractor of the police department, you have been asked to providea program capable of determining the required number of operators. If theprogram does not work correctly, you will be put in jail as well, along with allthe unhappy complainers. Do you really want to end up there?Telephone operators. Photo: SeattleMunicipal Archives.

Input

Each test case starts with two integers denoting the number of phone calls N (1<=N<=10 000) and thenumber of intervals M (1<=M<=100). This is followed by N lines describing the telephone calls, each oneconsisting of four integers Source, Destination, Start and Duration. Source and Destination identify thepair of telephone numbers establishing the connection (0<=Source;Destination<=10 000 000). Start andDuration are the start time and duration of the call in seconds (1<=Duration<=10000 and Start>=0).You can safely assume that the sum of Start and Duration ts into a 32-bit signed integer.Afterwards follow M lines containing the time intervals the police are interested in, each on described bytwo integers Start and Duration, in the same format and with the same meaning and constraints as thosein the telephone calls. The last test case is represented by N = M = 0 and must not be processed.

Output

For each of the M intervals of each test case, print the number of calls that are active during at least onesecond of the interval.

Sample input:

3 2

3 4 2 5

1 2 0 10

6 5 5 8

0 6

8 2

1 2

8 9 0 10

9 1

10 1

0 0

Sample output:

3

2

1

0

Added by:	Pfifing
Date:	2010-08-14
Time limit:	1s
Source limit:	50000B
Languages:	All except: PERL 6 TECS
Resource:	SWERC 2009

SPOJ Problem Set (tutorial)

7431. Vikram & Betaal

Problem code: RANJAN03

In order to pass through the wood, Vikram have to solve the following puzzle of Betaal:
Betaal has captured the Vikram' family in the location pointed by the co-ordinate (x, y), while Vikram is on (0,0). Now Vikram has to count the total no. of increasing lattice paths from his initial position to his family and tell it to Betaal in order to set his family free. As Vikram is a loser in mathematics from childhood, he demanded you to solve this puzzle and message it secretely to him.

Problem:

Given the no. of test cases, t, followed by t lines. Each line contain a no., n, representing the co-ordinate (n, n), where Vikram' family has been imprisioned. Find the number of increasing lattice paths from (0, 0) to (n, n) and print in separate lines. A path is increasing if it goes up or to the right only, i.e., a path can go from (x, y) to either (x+1, y) or (x, y+1).
Constraint:

a. 0 < n < 31
b. number of ways < 2^64

Example

`Input:``512345`**`Output:`**`262070252`

Added by: lost
Date: 2010-09-28
Time limit: 2s
Source limit:50000B
Languages: All
Resource: IIITM Local Contest

SPOJ Problem Set (tutorial)

7677. Frequent Values

Problem code: CPCRC1D

Given a non-decreasing series of integers $a_1, a_2, ..., a_n$ and indices 1<=i<=j<=n, what is the maximum number of repeated numbers within $a_i, a_{i+1}, ..., a_j$?

Input

Input contains several test cases.

Each case begins with two integers $1<=n,q<=10^5$.

Next line contains n integers ($a_1, a_2, ..., a_n$), each one having a size of lower than or equal to 10^5.

In next q lines, there are queries. Each one contains two integers 1<=i<=j<=n.

Input terminates when n,q are zero.

Output

For each query, print the maximum number of repetitions within numbers $a_i, a_{i+1}, ..., a_j$.

Example

Input:

```
10 31 1 3 3 3 3 5 10 10 102 31 105 100 0
```

Output:

```
143
```

Added by: Tii
Date: 2010-10-25
Time limit: 1s-3s
Source limit:50000B
Languages: All
Resource: ULM Local contest 2007

SPOJ Problem Set (tutorial)

7910. PECOS STRING

Problem code: PECOSSTR

As many of you are not aware of the Pecos, let me introduce him. *Pecos is the Jerry's guitar playing Uncle with a henkerin's for Cat Whiskers.*

Whenever a string of his guitar broke, he replace it with one the Tom's whiskers. Finding no other solution, Tom asks Uncle Pecos to let him play the music for him.

As Uncle Pecos is a creative mouse since his last life, he provide Tom his music notes, and ask him to play notes only at n^{th} position. On noticing that it's very easy, he added that the Tom had to miss every n^{th} note, from the notes which he is supposed to play.

Input

There are one test case per line. Test cases ends with a '-1' (qoutes for clearity). Each test case contains a number, n, followed by the musical notes, charaters from [a-z] and is upto length of 1000, in the same line.

Output

For each test case, print the required note in separate line.

Example

```
Input:3 abcdefghijklm4 pqrstuvwxyzsm-1Output:cflswsExplanation for test case # 1: abcdefghijklmInitially Tom is supposed to play every note at 3*n (n e Z +) position : cfilBut after the second restriction, he had to miss every 3*n' (n' e Z +) note, so new notes are : cfl
```

Added by: lost
Date: 2010-11-18
Time limit: 1s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

7925. Fibonnaci Parity

Problem code: FIBPARIT

In the quest to take over the world, the Pinky falls from the table, upside down. Miracle!!! Now he is intelligent. and the conversation goes like:

Brains : Pinky, are you pondering what I'm pondering?
Pinky : I think so, what would be the remainder when the n^{th} fibonacci number is divided by k?

Help Brain, solving this mystery.

Statement : Given n and k, find the remainder when the n^{th} fibonacci number is divided by k.
Constraints :
$1 <= n <= 10^4$
$1 < k <= 10^5$

n^{th} fibonnaci numbers are defined by :

$fib_n = 1$ if n = 1 or n = 2
$= fib_{n-1} + fib_{n-2}$ for n > 2

Fibonacci series goes like : 1 1 2 3 5 8 13...

Input

The first line contains t, number of test cases. In following t lines, there are two space separated numbers, n k.

Output

For each test cases, print the solution to the Pinky's quest in new line.

Example

```
Input:55 24 310 44 511 12Output:10335
```

Added by: lost
Date: 2010-11-19
Time limit: 2s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

8191. Gold distribution

Problem code: GOLDCOIN

Ankul and Rohil have been awarded with N gold coins in a recent programming competition. Now they want
to divide these N coins. But there is a problem, weight of each coin is not equal.Both of them know that
dividing these coins optimally is an NP-Complete problem. So they have decided to put all the coins on a stack and
divide them in the following manner:
They both take their turn alternatively and in each turn atmost K coins from the top of stack can be taken.
Ankul always start first. Consider that both of them are infinitely intelligent so they will always take the
best possible move. you have to find the maxinum weight which Ankul and Rohil will be able to get.

Ankul and Rohil have been awarded with N gold coins in a recent programming competition. Now they wantto divide these N coins. But there is a problem, weight of each coin is not equal.Both of them know thatdividing these coins optimally is an NP-Complete problem. So they have decided to put all the coins on a stack anddivide them in the following manner:

They both take their turn alternatively and in each turn atmost K coins from the top of stack can be taken.Ankul always start first. Consider that both of them are infinitely intelligent so they will always take thebest possible move. you have to find the maximum weight which Ankul and Rohil will be able to get.

Input

First line of input contains T,(1<=T<=30) the number of test cases then T test cases follow.First line of each test cases contains two space separated integers N and K (1<=N<=10000 and 1<=K<=10).2nd line contains N space separated integers (w1, w2,...wn), wi if the weight of ith gold coins from the top.(0<=wi<=1000)

Output

For each test case print one line which contains 2 space separated integers A and R.A and R are the maximum weight of gold which Ankul and Rohil will be able to take respectively.

Example

```
Input:
2
3 2
1 2 3
```

```
10 4
2 1 0 1 3 9 2 0 1 5
```

Output:

```
3 3
14 10
```

Added by:	Mahesh Chandra Sharma
Date:	2011-01-20
Time limit:	2s
Source limit:	50000B
Languages:	All
Resource:	Own problem

SPOJ Problem Set (main)

8214. Problem 1

Problem code: NOVICE21

Given three Integers A, B, and N. Find how many integers between A and B(inclusive) are divisible by N. 0<=A<=B<=10^18, 1<=N<=10^18.

Input

First line contains T the number of test cases. Each of next T lines contains 3 integer A, B, N.

Output

For each test case print the answer in a new line.

Example

Input:`21 4 310 20 6`**Output:**`12`

Added by: Mahesh Chandra Sharma
Date: 2011-01-25
Time limit: 1s
Source limit:50000B
Languages: C C# C++ 4.0.0-8 C++ 4.3.2 C99 strict JAVA
Resource: NSIT Noivce contest #2

SPOJ Problem Set (tutorial)

8216. Problem 3

Problem code: NOVICE23

For a given positive integer N, you have to find how many prime numbers are there upto N.

Input

First line contains T the number of test cases, (about 1000). Each of next T lines contains 1 integer N (1<=N<=1000000).

Output

For each test case print the answer in a new line.

Example

```
Input:
2510Output:
34
```

Added by: Mahesh Chandra Sharma
Date: 2011-01-25
Time limit: 1s
Source limit:50000B
Languages: All
Resource: NSIT Noivce contest #2

SPOJ Problem Set (tutorial)

8218. Problem 4

Problem code: NOVICE24

Given an integer N you have to find smallest prime number which comes after N, means smalltest prime which is greater than N.

Input

First line contains T the number of test cases. Each of next T lines contain one integer N. 1<=N<=10^9

Output

For each test case print the answer in a new line.

Example

```
Input:
2521Output:
723
```

Added by: Mahesh Chandra Sharma
Date: 2011-01-26
Time limit: 1s
Source limit:50000B
Languages: All
Resource: NSIT Noivce contest #2

SPOJ Problem Set (tutorial)

8219. Problem 5

Problem code: NOVICE25

A natural number k is divisor of another natural number if K completely divides N, means N % k = 0. For example 6 has 4 positive divisors 1, 2, 3, and 6. Now given a natural number N you have to find number of its positive divisors.

Input

First line contains T the number of test cases. Each of next T lines contain one integer N. 1<=N<=10^9

Output

For each test case print the answer in a new line.

Example

```
Input:
267Output:
42
```

Added by: Mahesh Chandra Sharma
Date: 2011-01-26
Time limit: 1s
Source limit:50000B
Languages: All
Resource: NSIT Noivce contest #2

SPOJ Problem Set (tutorial)

8289. Find the unique Number

Problem code: REPEAT

There will be odd number of element in array, and each number has a pair except one element. Find that unique element

Input

T = number of test cases

N = size of array

a1, a2, ... an array element

Output

Unique number

Example

```
Input:
252 1 2 3 173 3 5 4 5 4 5Output:
35 limit:T<=100N<=10^5ai <= 10^10
```

Added by:	pankaj
Date:	2011-02-10
Time limit:	1s-2s
Source limit:	50000B
Languages:	All
Resource:	own

SPOJ Problem Set (tutorial)

8293. Bit count

Problem code: BITCNT

Count the number of ones in bit repesentation of number.

Generate N number using formula (a*i+b)%c for all 1<=i<=n

limit: 1<=N<=1000000

0<=a*N, b*N<2^64

1<=c<2^64

Input

N

a b c

Output

[N numbers]

c1

c2

..

cn

Example

```
Input:
55 6 7Output:
11022ExplanationGenerated numbers are 4, 2, 0, 5, 3 and bit counts are 1, 1, 0, 2, 2 respectivelyhints:use unsigned long long
```

Added by:	pankaj
Date:	2011-02-10
Time limit:	1s-2s
Source limit:	50000B
Languages:	All
Resource:	own

SPOJ Problem Set (classical)

8353. GUUGU

Problem code: GUUGU

Guuugu

3/5 bolčok bar dejli.Bul bolčok guuugu alfavitinde(birok kyrgyzča soz) myndaj žazylat GUZSB AWD(že myndaj ABCGU GIR==učton beš).
Silerge guuugu alfavitinde berilgen bolčoktu kajra san turundo bolčoko ajlandyrgyla bolušunča maanisi minimum bolsun.
(0<Minimum maani<1)

Example

`Input:``ABCGU GIR`**`Output:`**`3/5`

minimum maani 1 bolušun mumkun

Added by:	Emil Sulaymanov
Date:	2011-02-16
Time limit:	5s
Source limit:	50000B
Languages:	All
Resource:	PANKOV PROBLEM

SPOJ Problem Set (tutorial)

8360. Prime After N

Problem code: AU12

Given an integer N you have to find smallest prime number which comes after N, means smalltest prime which is greater than N.

Input

First line contains T(1<=T<=1000) the number of test cases. Each of next T lines contain one integer N. 1<=N<=10^9

Output

For each test case print the answer in a new line.

Example

Input:`2521`

Output:

```
723
```

Added by: amit karmakar
Date: 2011-02-17
Time limit: 3s
Source limit:50000B
Languages: All

SPOJ Problem Set (classical)

8378. find seat in dope cup

Problem code: DOPECUP

N friends go to watch the 2011 DOPE-cup. Unfortunately for them, they reach late to the venue and as a result, could not get the seats together. Suppose that the seats are arranged in R rows and C columns. Your job is to write a program to find seats as close to each other as possible. The nearness is defined as the area of the smallest rectangle with sides parallel to the seats that contains all seats for the group.Area is measured as the number of seats contained in it.

Input

Each test case will consist on several lines.
The first line will contain three positive integers R, C and N as explained above
($1 <= R,C <= 300$, $1 <= N <= R \times C$).
The next R lines will contain exactly C characters each. The j-th character of the i-th line
will be 'X' if the j-th seat on the i-th row is taken or '.' if it is available.
There will always be at least N available seats in total. input ends with R,C,N is 0 0 0.

Output

Area is measured as the number of seats contained in it.

Example

Input:
```
4 3 3
..X
X.X
...
XXX
5 4 6
XX..
..X.
XX..
...X

XXXX
0 0 0
```
Output:
```
3
8
```

Added by:	pankaj
Date:	2011-02-19
Time limit:	5s-15s
Source limit:	50000B
Languages:	All
Resource:	FSEAT, DOPE 2011

SPOJ Problem Set (main)

8379. count frequency of digits

Problem code: DOPECNT

Young Dope was bored of finding whether a given number is palindromic or not.So he started another exercise described as follows. Given a number consisting of n digits, find the number of pairs of digits such that position[i] equals position[j] 1<=i,j<=n.

Input

First line contains T, the number of test cases <100
Each test case contains a number with 1=<length <= 10^5 and
digits only between 0 and 9 both inclusive.

Output

Number of pairs of such digits.

Example

Input:

2
1234
777

Output:

4
9

Added by: pankaj
Date: 2011-02-19
Time limit: 2s-3s
Source limit:50000B
Languages: C C++ 4.0.0-8 C++ 4.3.2
Resource: harsh, DOPE 2011

SPOJ Problem Set (tutorial)

8395. ZBROJ

Problem code: ZBROJ

After he got tired of rotating tables, the teacher gave Perica another problem. She wrote two integers, A
and B, on the blackboard and asked him to add them.
Perica is never wrong with his calculation, but sometimes he doesn't copy the numbers correctly. The only mistake he ever makes is copying a 5 as a 6, and viceversa.
Given two numbers, A and B, calculate the minimum and the maximum sum Perica could possibly get.

Input

The first and only line of input contains positive integers A and B (1 <= A, B <= 1 000 000).

Output

In single line of output, print two space separated integers, minimum and maximum sum Perica could get.

Example

Input:`11 25`**Output:**`36 37`

Added by: akaki
Date: 2011-02-22
Time limit: 1s
Source limit:50000B
Languages: All
Resource: coci

SPOJ Problem Set (tutorial)

8412. Greedy Government

Problem code: GRDGOVT

The Government of Greedistan has just found out that there is a lot of gold beneath some land area in their country. They quickly surveyed the land area and found that some people are living there, and surprisingly the houses are organized in a **M** x **N** grid, each cell having exactly one house.

To acquire all the gold beneath this area, the government has planned to make up a rumor (a volcano is about to erupt near by) and let the people vacate their houses themselves. They start the rumor at only one house on day 1. It takes exactly one day for the rumor to spread from a house to any of its neighbors (top, left, bottom, right). They all vacate their houses as soon as they know about the volcano.

The government wants to choose the starting house for the rumor in such a way that it takes minimum number of days for all the people to vacate their houses. Find this minimum time.

Input

The first line contains **T**, the number of test cases. Each of the next **T** lines contain two integers **M** and **N**. If one of M or N is zero, other one will also be zero, and this means, there are no houses at all.

T <= 1,000

0 <= **M** <= 1,000,000

0 <= **N** <= 1,000,000

Output

The minimum number of days it takes to spread the rumor to all houses.

Example

3	2	3	4
2	1	2	3
3	2	3	4

Added by:	Anil Kishore
Date:	2011-02-25
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	CodeMutants 2011 , DA-IICT, India

SPOJ Problem Set (tutorial)

8413. Clean the Forum Posts

Problem code: CLNFORUM

Long ago during the Jurassic Park age, there was a company TopHunters, which used to conduct hunting competitions for the Dinosaurs around the world. Petrosaurus was believed to be the greatest hunter of that time and all other dinosaurs enjoyed watching the hunting videos posted by him in the forums (yes, they had a website :)).

Many Indian dinosaurs used to post in sms language in the forums and annoy others. Vexosaurus was so annoyed reading the posts with numbers used in place of letters. He decided to correct all those posts, before the forums get *Rusty*. Could you please code for him and replace the following words in 1st column with their corresponding words in 2nd column

8	ate
w8	wait
gr8	great
4	for
b4	before

Lets not make this complex and just replace the words when they appear entirely as a word themselves, and not as a subpart of bigger words (see examples for clarity). A word is a continuous sequence of non-space characters.

Disclaimer : This is just a fictional story. Any resemblance to real persons or company or dinosaurs is purely coincidental :|

Input

First line contains T [number of test cases, around 50]. Each of the next T lines contains a sentence with not more than 100 characters ['a' - 'z' , '0' - '9' , space]. There can be 2 or more continuous spaces and you have to preserve them, in the output. Input sentence will not begin or end with a space.

Output

For each test case, output the corrected sentence, in a new line.

Example

```
Input:3i 8 food b4gr8 2 see you w8ing1234   5678   9Output:i ate food beforegreat 2 see you w8ing1234   5678   9
```

Added by: Anil Kishore
Date: 2011-02-25
Time limit: 1s
Source limit:50000B
Languages: All
Resource: CodeMutants 2011 , DA-IICT, India

SPOJ Problem Set (tutorial)

8437. MODULUS

Problem code: PROBLEM1

Problem1: MODULUS

Problem

Given an integer n print all the possibilities of a%n where a can be any positive integer.

Input

The first line consists of an integer t, the number of test cases followed by t lines containing an integer n.

Output

For each test case print all the possibilities of a%n in descending order separated by a single space. After each test case print a new line character. If there are no possibilities print "NOT POSSIBLE\n".

Input specifications:

0<t<=100

0<=N<=100

Time limit: 1 second

Example

Sample Input

2

1

2

Sample Output

0

1 0

Added by: cegprakash
Date: 2011-03-01
Time limit: 1s
Source limit:250B
Languages: All

SPOJ Problem Set (tutorial)

8440. Problem 1

Problem code: NOVICE41

On a board on size N*N. Johar has placed N stones such that in every row and every column there is exactly one stone. Moreover in every diagonal and anti diagonal there is at most one stone.Now Kandarp want to check his solution because he does not trust Johar. So he want you to check whether he has placed these N stones correctly or not.

Input

First line contains T, the number of test cases. then T test cases follow. First line of each test case contains N (1<=N<=50) then each of next N lines contains an string of N characters. jth character of ith string is '#' is there is an stone at position (i,j) otherwise it is '.'.

Output

For each test case print YES if it is a valid arrangement or NO if it is invalid.

Example

```
Input:
22
3
..#
#..
.#.
4
.#..
...#
#...
..#.

Output:
NO
YES
```

Added by: Mahesh Chandra Sharma
Date: 2011-03-01
Time limit: 1s
Source limit:50000B
Languages: All
Resource: Own problem

SPOJ Problem Set (tutorial)

8441. Problem 2

Problem code: NOVICE42

This contest is based on brute force, and where better to apply this technique than in a day to day newspaper game. Hemanshu Bansal has a knack for solving puzzles and he claims that he is very fast always saying that he can solve the problem even before I can start to code. Help me beat him once in for all in this famous game of Sudoku.
The objective of Su Doku puzzles is to replace the blanks in a 9 by 9 grid in such that each row, column, and 3 by 3 box contains each of the digits 1 to 9.
You will be given a Sudoku puzzle and your program has to print its solution.

This contest is based on brute force, and where better to apply this technique than in a day to day newspaper game. Hemanshu Bansal has a knack for solving puzzles and he claims that he is very fast always saying that he can solve the problem even before I can start to code. Help me beat him once in for all in this famous game of Sudoku.The objective of Su Doku puzzles is to replace the blanks in a 9 by 9 grid in such that each row, column, and 3 by 3 box contains each of the digits 1 to 9.

You will be given a Sudoku puzzle and your program has to print its solution.

Input

line 1:T(no. of test cases)

line 2: Grid 01

line 3-11: grid of 9x9

line 12: Grid 02

line 13-21: grid of 9x9

....

....

so on.

Output

line 1: Grid 01(should be same as input)

line 2-10: grid of 9x9(the solution)

line 11: <blank line>

line 12: Grid 02

....

so on.

In case of multiple solutions print lexicographically minimum solution. Refer to wikipedia for the definition of lexicographical order.

Example

Input:
```
2
Grid 01
003020600
900305001
001806400
008102900
700000008
006708200
002609500
800203009
005010300
Grid 02
200080300
060070084
030500209
000105408
000000000
402706000
301007040
720040060
004010003
```

Output:
```
Grid 01
483921657
967345821
251876493
548132976
729564138
136798245
372689514
814253769
```

```
695417382

Grid 02
245981376
169273584
837564219
976125438
513498627
482736951
391657842
728349165
654812793
```

Added by: Mahesh Chandra Sharma
Date: 2011-03-01
Time limit: 20s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

8443. Problem 4

Problem code: NOVICE44

Piyush is a very inteligent chap, he has a facination for maths and is never convinced without proof of anything. Last time I told him that sqrt(2) can be written as an expansion of a series as sqrt(2) = 1 + 1/(2 + 1/(2 + 1/(2 + ...))) = 1.414213...
Now i need to prove this to him. Being a fan of finding all solutions i have descided to use a program to find all possible fractions that can be formed using this series with depth = N and show it to piyush, I need your help to do this.
example:
N=1 : 1 + 1/2 = 3/2
N=2 : 1 + 1/(2 + 1/2) = 7/5
N=3 : 1 + 1/(2 + 1/(2 + 1/2)) = 17/12
and so on...
Given a value of N(<=40) print the fraction in lowest form. Lowest form means that GCD(numerator,denominator) = 1

Piyush is a very inteligent chap, he has a facination for maths and is never convinced without proof of anything. Last time I told him that sqrt(2) can be written as an expansion of a series as sqrt(2) = 1 + 1/(2 + 1/(2 + 1/(2 + ...))) = 1.414213...

Now I need to prove this to him. Being a fan of finding all solutions I have descided to use a program to find all possible fractions that can be formed using this series with depth = N and show it to piyush, I need your help to do this.

example:

N=1 : 1 + 1/2 = 3/2

N=2 : 1 + 1/(2 + 1/2) = 7/5

N=3 : 1 + 1/(2 + 1/(2 + 1/2)) = 17/12

and so on...

Given a value of N(<=40) print the fraction in lowest form. Lowest form means that GCD(numerator,denominator) = 1

Input

line 1: T(number of test cases)

line 2 to T+1: vaue of N for each test case

Output

numerator/denominator in the lowest form for each test case

Example

```
Input:
4
1
2
3
4

Output:
3/2
7/5
17/12
41/29
```

Added by:	Mahesh Chandra Sharma
Date:	2011-03-01
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	Own problem

SPOJ Problem Set (tutorial)

8444. Problem 5

Problem code: NOVICE45

Mahesh and I are always at conflict, Where I am a lazy person who would rather apply bruteforce and compute all possibilities of a problem to find the answer, he would rather solve it on paper to find a mathematical formula and reduce computational time. Fed up of his daily taunts I formulated the following problem, lets see who wins here bruteforce or mathematical precalculation:
Given two positive integers N and K you have to find the number of distinct ways of writing N as the sum of integers(possibly only 1) in range of [1,K] (inclusive).
For example if N = 4 and K = 2, we have these 3 ways => (2+2), (2+1+1), (1+1+1+1).

Mahesh and I are always at conflict, Where I am a lazy person who would rather apply bruteforce and compute all possibilities of a problem to find the answer, he would rather solve it on paper to find a mathematical formula and reduce computational time. Fed up of his daily taunts I formulated the following problem, lets see who wins here bruteforce or mathematical precalculation:

Given two positive integers N and K you have to find the number of distinct ways of writing N as the sum of integers(possibly only 1) in range of [1,K] (inclusive).

For example if N = 4 and K = 2, we have these 3 ways => (2+2), (2+1+1), (1+1+1+1).

Input

Line 1: T(number of test cases)

Line 2 to T+1: 2 space seprated integers N,K. (1<=N<=10000 and 1<=K<=100)

Output

1 line per test case telling the number of ways. Since the answer can be very large print it modulo 1000000007.

Example

```
Input:
3
1 10
2 1
4 2

Output:
1
1
3
```

Added by:	Mahesh Chandra Sharma
Date:	2011-03-01
Time limit:	2s
Source limit:	50000B
Languages:	All
Resource:	Own problem

SPOJ Problem Set (tutorial)

8450. C ONE

Problem code: C1

Write a C program to print "Hello World"

Output

Hello World

Example

`Output:``Hello World`

Added by:	styrofox
Date:	2011-03-03
Time limit:	1s
Source limit:	50000B
Languages:	C
Resource:	C

SPOJ Problem Set (tutorial)

8455. C TWO

Problem code: CTWO

Write a C program to accept two consecutive numbers from the user and print their sum.

Input

The two numbers entered by the user.

Output

The sum of the numbers.

Example

```
Test Case 0Input:
1 2 Output:
3Test Case 1Input:2 3Output:5Test Case 2Input:3 4Output:7Test Case 3Input:5 6Output11
```

Added by: styrofox
Date: 2011-03-04
Time limit: 1s
Source limit:50000B
Languages: C C99 strict
Resource: C

SPOJ Problem Set (tutorial)

8459. AVERYEASYPROBLEM

Problem code: PROBLEM6

PROBLEM6: AVERYEASYPROBLEM

Problem Statement:

Given two numbers a and b find the value of a!/b!

Input:

The first line consists of an integer t, the number of test cases. Then the next t lines consists of two integers a and b.

Output:

For each test case print the value of factorial(a)/ factorial(b) provided b is always less than a.

Input Constraints:

1<=t<=100

0<=b<=a<=15

Example

Sample Input:

3

15 0

13 2

1 0

Sample Output:

1307674368000

3113510400

1

Added by: cegprakash
Date: 2011-03-04
Time limit: 2s
Source limit:200B
Languages: C C++ 4.0.0-8 C++ 4.3.2

SPOJ Problem Set (classical)

8465. C THREE

Problem code: C3

Write a C program to accept any two numbers from the user and print their sum using a function.

Input

The two numbers entered by the user.

Output

The sum of the numbers calculated by the function

Example

Test Case 0

```
Input:
1 3
Output:
4
```

Added by: styrofox
Date: 2011-03-05
Time limit: 1s
Source limit:50000B
Languages: C C99 strict
Resource: C

SPOJ Problem Set (tutorial)

8468. AVERAGE

Problem code: AVG

Problem: AVERAGE

Problem Statement:

Given n numbers find the average of them.

Input:

The first line consists of an integer t, the number of test cases. Each test case consists of 2 lines. In each test case the first line consists of an integer n, the count of numbers. The next line consists of n integers.

Output:

For each test case print the average of the numbers without decimal places

Sample Input:

2

5

1 2 3 4 5

3

2 1 1

Sample Output:

3

1

Added by:	cegprakash
Date:	2011-03-06
Time limit:	1s
Source limit:	50000B
Languages:	C C++ 4.0.0-8 C++ 4.3.2 JAVA

SPOJ Problem Set (tutorial)

8492. Intersecting Circles

Problem code: CIRCINT

Given two circles with centers at (x1, y1) and (x2,y2) and having radius r1 and r2 respectively, find if they intersect or not. If two circles touch then they are considered to be intersecting.

Given two circles with centers at (x1, y1) and (x2,y2) and having radius r1 and r2 respectively, find if they intersect or not. Two circles are considered to be intersecting if they have a common area. Even if two circles touch at a point then they are considered to be intersecting.

Input

First line contains an integer T. Then follow T lines each line containing integers x1, y1, x2, y2, r1, r2 in that order.

Constraints:
T <= 10,000
All other integers will have an absolute value <= 1000,000,000

Output

Print "YES" (without quotes) if they intersect and "NO" if they don't intersect.

Example

```
Input:
3
0 0 2 2 1 1
0 0 2 2 3 3
0 0 1000000000 0 600000000 400000000

Output:
NO
YES
YES
```

Added by:	.:: Pratik ::.
Date:	2011-03-07
Time limit:	10s
Source limit:	50000B
Languages:	All

SPOJ Problem Set (tutorial)

8493. Colorful Blocks

Problem code: COLORF

Rith, the student of the month has received k sets of colored blocks. Each set is of a different color and each block in a set is identical to any other block. Rith has n-types of colors and has a1, a2, a3 ... an number of blocks for each color respectively. He arranges these blocks in a straight line, and wants to know the number of ways he can arrange it. Please help him find out the number of ways. Oh and Rith knows that this number will be very large, and hence asks you to find it modulo 1,000,000,009 (A number modulo P is the remainder that is left after dividing that number by P)
For example if Rith has 2 types of colors and {a1,a2} = {1,2}
Then the following arrangemenets are possible
(Digit here means the color)
122
212
221
Hence the answer is 3.

Rith, the student of the month has received k sets of colored blocks. Each set is of a different color and each block in a set is identical to any other block. Rith has n-types of colors and has a1, a2, a3 ... an number of blocks for each color respectively. He arranges these blocks in a straight line, and wants to know the number of ways he can arrange it. Please help him find out the number of ways. Oh and Rith knows that this number will be very large, and hence asks you to find it modulo 1,000,000,009 (A number modulo P is the remainder that is left after dividing that number by P)

For example if Rith has 2 types of colors and {a1,a2} = {1,2}

Then the following arrangemenets are possible

(Digit here means the color)

122

212

221

Hence the answer is 3.

Input

The first line contains an integer T, which is the number of test cases. Then there are T-test case blocks which follow.

Each test-case block starts with an integer n, which is the number of types of colors.

The next line contains n-integers a1, a2, a3 ... an as described in the statement.

T <= 100

1 <= n <= 100

a1 + a2 + a3 .. + an <= 200,000

Output

Print the number of ways as described modulo 1,000,000,009

Example

```
Input:
3
2
1 2
3
1 1 1
5
200 200 200 200 200


Output:
3
6
706392408
```

Added by:	.:: Pratik ::.
Date:	2011-03-07
Time limit:	12s
Source limit:	50000B
Languages:	All

SPOJ Problem Set (tutorial)

8494. Lucky Numbers

Problem code: LUCKYN

Shrek and Kung Fu Panda once met after having no forthcoming prequels. They quickly noticed that both of them were superstitious and this helped them bond a lot.
Shrek believes that the number 4 is lucky and Kung Fu Panda believes that number 7 is lucky. You being their friend want to list down numbers in increasing order that consist only of 4 or 7.
The first few elements of the list are 4, 7, 44, 47, 74, 77, 444 ... You must answer the n-th (1-based, 4 is the 1st term of the sequence)

Shrek and Kung Fu Panda once met after having no forthcoming prequels. They quickly noticed that both of them were superstitious and this helped them bond a lot.

Shrek believes that the number 4 is lucky and Kung Fu Panda believes that number 7 is lucky. You being their friend want to list down numbers in increasing order that consist only of 4 or 7.

The first few elements of the list are 4, 7, 44, 47, 74, 77, 444 ... You must answer the n-th (1-based, 4 is the 1st term of the sequence)

Input

The first line contains the number of test-cases T

The following T-lines contains an integer n.

T <= 10,000

n <= 1000,000,000

Output

Print the n-th term in the sequence of lucky numbers

Example

```
Input:
6
1
20
300
4000
50000
1000000000

Output:
4
4747
44747747
77774744447
744447747474447
7747774477474774474744444447
```

Added by: .:: Pratik ::.
Date: 2011-03-07
Time limit: 8s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

8790. Find summits

Problem code: SUMMIT

Given an altitudinalmap(0 <= altitudes <= 100000) find the summits i.e. all points which are bigger than alltheirneighbours. There is at least one summit in each map.

Score is source length.

Input

The number n ofmaps (n<=50) in the first line.
Then for each map one line withits width w and heigth h (3<=w,h<=20) separated by a space. After this the h rows ofthe map.

Output

The space-separated summits in ascending order.

Example

```
Input:
23 371034 8558 6594118265 1226 7407671003 75481 284463 322360 72964 4789175416 7746 8043295606 4613 83341

Output:
71034 7548183341 95606
```

Added by: HWK
Date: 2011-05-02
Time limit: 10s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

8865. Brackets

Problem code: BRKTS

Your task is to determine if a sequence of opening and closing brackets is a valid expression. A valid expression is one where every opening bracket has a corresponding closing bracket.

Input

The input consists of several test cases. The first line of the input is an integer 0< n < 500. n lines follow, each one consisting of a sequence of brackets, which represents an expression.

Output

For each sequence in the input, print ":)" if the sequence is a valid expression, and ":(" otherwise.

Example

```
Input:
4{}}}{{{{{{{}{}}{}{{}}}}}Output:
:):(:):(
```

Added by: Simón Santiago Soriano Pérez
Date: 2011-05-12
Time limit: 1s
Source limit: 50000B
Languages: C++ 4.3.2 JAVA
Resource: https://docs.google.com/document/d/1WYMA9kda4g9zH7in5AsHjctKvkx8Nowc-V6FbMIpM_k/edit?hl=en&authkey=CMjAyJQO#

SPOJ Problem Set (oi)

9015. FARMER

Problem code: LQDFARM

A farmer has a set of fields, each of which is surrounded by cypress trees. Also, the farmer has a set of strips of land, each of which has a row of cypress trees. In both fields and strips, between every two consecutive cypress trees is a single olive tree. All of the farmer's cypress trees either surround a field or are in a strip and all of the farmer's olive trees are between two consecutive cypress trees in a field or in a strip.

One day the farmer became very ill and he felt that he was going to die. A few days before he passed away he called his eldest son and told him, "I give you any *Q* cypress trees of your choice and all the olive trees which are between any two consecutive cypress trees you have chosen." >From each field and from each strip the son can pick any combination of cypress trees. Since the eldest son loves olives he wants to pick the *Q* cypress trees which will allow him to inherit as many olive trees as possible.

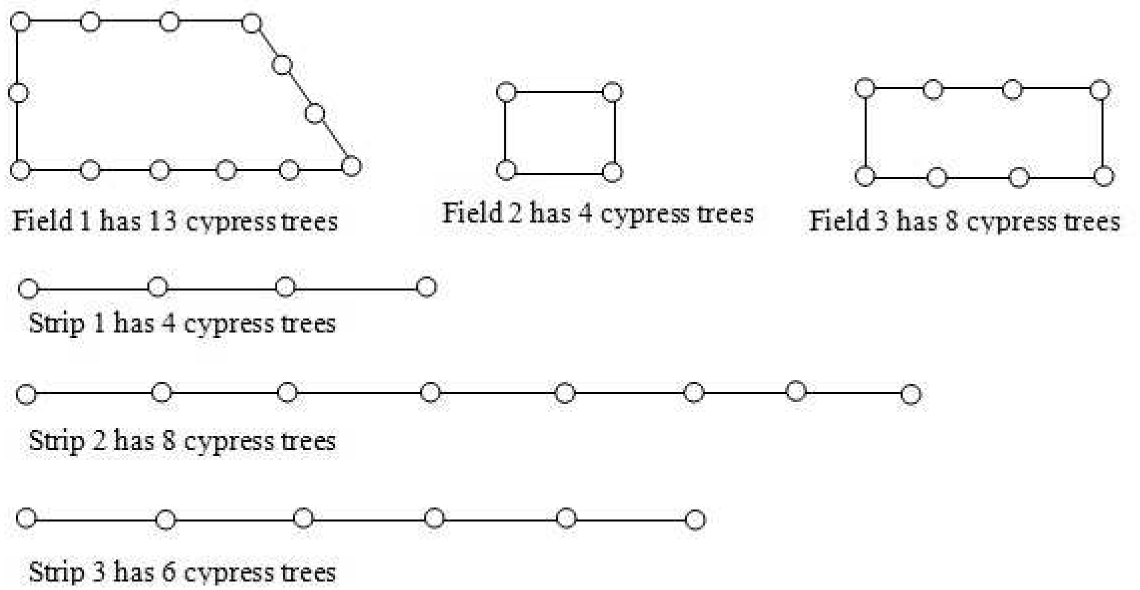


Figure 1 An example setting of cypress trees; olive trees are not shown.

In Figure 1, assume that the son is given *Q*=17 cypress trees. To maximize his olive inheritance he should choose all the cypress trees in Field 1 and Field 2, inheriting 17 olive trees.

You are to write a program which, given the information about the fields and the strips and the number of cypress trees the son can pick, determines the largest possible number of olive trees the son may inherit.

INPUT

The input file name is farmer.in. The first line contains first the integer *Q*: the number

of cypress trees the son is to select; then the integer *M*, the number of fields; and then the integer *K*, the number of strips. The second line contains *M* integers *N1*, *N2*,... *NM*, : the numbers of cypress trees in fields. The third line contains *K* integers *R1*, *R2*,... *RK*: the numbers of cypress trees in strips.(0<=Q<=150000, 3<= Ni<=150, 2<= Ri<= 150)

OUTPUT

The output file name is farmer.out. The file is to contain one line with one integer: the largest possible number of olive trees the son may inherit.

EXAMPLE INPUTS AND OUTPUTS

Input **Output**

17 3 3 17

13 4 8

4 8 6

Added by: qu-t!-r?poO
Date: 2011-06-11
Time limit: 0.300s-1.100s
Source limit:50000B
Languages: All except: CLOJ ERL F# GO PERL 6 PYTH 3.1.2 SCALA TCL
Resource: IOI 04

SPOJ Problem Set (classical)

9025. String Search

Problem code: STRINGS

A rectangular board with R rows and C columns is given. Each cell of the board contains an alphabet.Now Q queries are asked. In each query a word is provided and you have to identify the word in the board. A word can be made by continuous alphabet cells on board in any direction. It is not neccesary to follow a particular direction only... What this means is that cells may not lie on a straight line and adjacent alphabets in given word must also be adjacent on the board.Each cell is adjacent to all its neibhour cells (diagonally also).Also each cell is counted only once. You have to tell starting and ending cell co-ordinates on the board for the given string. All coordinates in this problem are 0-indexed and in the form (row, column).

Input

First line contains integers R and C ,number of rows and columns in the board.Following R lines with C characters describes the board. Then next line contains an integer Q number of queries. Q lines follows. Each line contains a string of alphabets.Input is such that it is possible to get a unique solution within time contraints.

0<R,C<=20

0<Q<=30

String length<R*C.

Output

For each query output the starting and ending co-ordinates of the cells forming the string on the board in the format (x1,y1) (x2,y2) ,where (x1,y1) and (x2,y2) are co-ordinates of the starting cell and ending cell respectively.

If there are more than one solution then output that points for which distance between them is minimum.

Example

```
Input:
4 44 4
acgt
tdtd
fcyy
ithc
2
catch
if
acgttdtd
fcyyInput:Input:Output:
```

```
(0,1) (3,2)
(3,0) (2,0)
```

Added by:	Manohar Singh
Date:	2011-06-13
Time limit:	5s
Source limit:	50000B
Languages:	All
Resource:	Manohar Singh

SPOJ Problem Set (classical)

9044. Magic Square!

Problem code: MAGICSQU

Given an nxn matrix, if it satisfy the next constraints it could be called a "Magic Square".

Constraints:

* The sum of the n rows, n columns and the two main diagonals of the matrix must be equal to the same number m, called magic constant
* the n^2 positions of the matrix must contains natural numbers between 1 to n^2, with no repetitions.

Input

Each test case consist in a natural number n (0 < n <= 20) denoting the matrix dimension. Then, the next n^2 natural numbers will represent the values that initially contains the matrix.
For each position a_{ij} (1 <= i,j <= n) of the matrix, we have: 0 <= a_{ij} <= n^2, for all ij.
A value of 0 in an initial position aij of the matrix means that you must find a number to complete that place. A non-zero value means that you cannot change that value for the given position.

Output

For each case you must print the case number (starting from 1).
Then, in the next n lines you must print the magic square if exists or the phrase ’’No Solution’’ (without the quotes) if doesn’t exist a solution for the given initial configuration.
See the output example.
If for a given initial configuration exists more than one solution, you should give the lower one assuming a lexicographic order. For lexicographic comparison you should consider lines in first place.
Print a blank line between test cases.

For example:
2 7 6
9 5 1
4 3 8
it’s lexicographically before than:
4 3 8
9 5 1
2 7 6

Example

Input: `38 0 00 0 00 0 021 10 0`

Output:

`Case #18 1 6 3 5 7 4 9 2 Case #2No Solution`

Added by:	Coach UTN FRSF
Date:	2011-06-17
Time limit:	3s
Source limit:	50000B
Languages:	All

SPOJ Problem Set (tutorial)

9078. Graph basics concepts

Problem code: GRAPHS11

Description:

In this problem, your program will have to read a weighted graph (directed or non directed), and perform some basic algorithms on the graph:

- Visualization of the graph (in the same format of the input)

- BFS traversal and show the paths from the node 1 (the root node), to the rest of the nodes.

- DFS traversal and show the paths from the node 1 (the root node), to the rest of the nodes.

- Visualization of connected components

- Visualization of bridges

Input

The graph to be read comes in several lines: in the first line, you will have a value *d*: 0 (if the graph is a non directed graph), or 1 (if it is a directed one).

In the second line you will have a pair of numbers: *n* and *e*, the first is the quantity of nodes in the graph, and the second is the quantity of edges in the graph.

Then, you will have *e* lines, with triplets: *o*, *t* and *w*, where *o* is the source node of an edge, *t* is the target node, and *w* is the weight.

The triplets are ordered from the edges with origin at the root node (number 1), in an ascending way. Also, if there are more than one edge starting from one node, they are ordered by target node.

The numbers of the nodes are consecutive: 1, 2, ...*n*

Output

You will have to show all the information in the list showed above (in the **Description**) section. Take a look at the Example below to see the format to follow.

Example

```
Input:07 81 2 11 3 12 4 13 4 14 5 14 6 15 7 16 7 1Output:07 81 2 11 3 12 4 13 4 14 5 14 6 15 7 16 7 1BFS: 1 2 3 4 5 6 7BFS Paths:11 21 31 2 41 2 4 51 2 4 61 2 4 5 7DFS:1 2 4 3 5 7 6DFS Paths:11 21 2 4 31 2 41 2 4 51 2 4 5 7 61 2 4 5 7Connected Components:C1: 1 2 3 4 5 6 7Bridges:0
```

Added by:	Coach UTN FRSF
Date:	2011-06-27
Time limit:	1s-2s
Source limit:	50000B
Languages:	All

SPOJ Problem Set (tutorial)

9082. Graph basics concepts

Problem code: GRAPHS12

Description:

In this problem, your program will have to read a weighted graph (directed or non directed), and perform some basic algorithms on the graph:

- Visualization of the graph (in the same format of the input)

- BFS traversal and show the paths from the node 1 (the root node), to the rest of the nodes.

- DFS traversal and show the paths from the node 1 (the root node), to the rest of the nodes.

- Visualization of connected components

- Visualization of articulation vertices (the numbers of the vertices that are articulation points)

Input

The graph to be read comes in several lines: in the first line, you will have a value *d*: 0 (if the graph is a non directed graph), or 1 (if it is a directed one).

In the second line you will have a pair of numbers: *n* and *e*, the first is the quantity of nodes in the graph, and the second is the quantity of edges in the graph.

Then, you will have *e* lines, with triplets: *o*, *t* and *w*, where *o* is the source node of an edge, *t* is the target node, and *w* is the weight.

The triplets are ordered from the edges with origin at the root node (number 1), in an ascending way. Also, if there are more than one edge starting from one node, they are ordered by target node.

The numbers of the nodes are consecutive: 1, 2, ...*n*

Output

You will have to show all the information in the list showed above (in the **Description**) section. Take a look at the Example below to see the format to follow.

Example

Input:`07 81 2 11 3 12 4 13 4 14 5 14 6 15 7 16 7 1`**Output:**`07 81 2 11 3 12 4 13 4 14 5 14 6 15 7 16 7 1BFS: 1 2 3 4 5 6 7BFS Paths:11 21 31 2 41 2 4 51 2 4 61 2 4 5 7DFS:1 2 4 3 5 7 6DFS Paths:11 21 2 4 31 2 41 2 4 51 2 4 5 7 61 2 4 5 7Connected Components:C1: 1 2 3 4 5 6 7Articulation Vertices:4`

Added by:	Coach UTN FRSF
Date:	2011-06-28
Time limit:	1s-2s
Source limit:	50000B
Languages:	All

SPOJ Problem Set (tutorial)

9106. REARRANGE

Problem code: NOVICE61

Adam's parents put up a sign that says "COMBINATORICS". The sign is so big that exactly one letter fits on each panel. Some of Adam's younger cousins got bored during the reception and decided to rearrange the panels. How many unique ways can the panels be arranged (counting the original arrangement)?

Input

First line contains a single integer T(less than 1000) denoting the total number of test cases.

Each test case contains a string S having atmost 15 characters from A to Z.

Output

For each word, output the number of unique ways that the letters can be rearranged (counting the original arrangement). Use the format shown in Sample Output, below.

Example

```
Input:
3
HAPPY
WEDDING
ADAM

Output:
Case 1: 60
Case 2: 2520
Case 3: 12
```

Added by: amit karmakar
Date: 2011-07-02
Time limit: 1s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

9108. SHAKE

Problem code: NOVICE66

Consider a meeting of n businessmen sitting around a circular table. To start the meeting, they must shake hands. We say that the shake is perfect if no arms cross each other and each businessman shakes the hand of exactly one other businessman. All handshakes happen simultaneously.

Input

First line contains an integer T(less than 5000) denoting the total number of test cases.

Each test case consists of a single integer n(1 <= n <= 5000) in one line.

Output

For each tese case, print the number of perfect shakes that exist for n businessmen modulo 1000000007.

Example

```
Input:
1

6

Output:

5
```

Added by: amit karmakar
Date: 2011-07-03
Time limit: 3s-4s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

9109. NTH WORD

Problem code: NOVICE67

It is possible to number all the words that could be formed from a specified alphabet. One way to do it is to put the words in order based on length, with words of common length ordered alphabetically. Using this scheme, if we use the alphabet consisting of just the 3 letters 'a', 'b', and 'c' then the words are numbered:

1:a 2:b 3:c 4:aa 5:ab 6:ac 7:ba 8:bb etc.

There are an infinite number of possible words, but each has its own positive number. We want to be able to find the word that corresponds to any given number **N**. The alphabet to be used is the first **A** letters of the normal lowercase alphabet, with their usual alphabetical ordering.

Input

First line contains an integer **T**(less than 1000) denoting the total number of test cases. Each test case consists of a single line of input having two integer **A**(2<=**A**<=26)and **N**(1<=**N**<=2000000000)**.**

Output

For each test case print the required string in a single line.

Example

```
Input:
2
3 13
26 2000000000

Output:
aaa
flhomvx
```

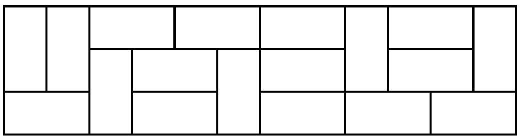
Added by: amit karmakar
Date: 2011-07-03
Time limit: 2s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

9110. TILE IT

Problem code: NOVICE68

In how many ways can you tile a 3xn rectangle with 2x1 dominoes? Here is a sample tiling of a 3x12 rectangle. Input consists of several test cases followed by a line containing -1. Each test case is a line containing an integer 0 <= n <= 30. For each test case, output one integer number giving the number of possible tilings.



SAMPLE INPUT

```
2
8
12
-1
```

SAMPLE OUTPUT

```
3
153
2131
```

Added by: amit karmakar
Date: 2011-07-03
Time limit: 1s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

9111. Prime Time

Problem code: NSUJ02A

Maria really loves prime times. In her definition, a time is called prime time, if both numbers in hour and minute are primes. For example 07:11 is a prime time, since both of them are prime numbers, but 04:17 or 07:09 are not prime time.
You decided to write a bot that will call her in prime times, now you have to write a program that can understand if a time is a prime time or not.

Input

First number in the input is t, number of test cases.
After that t lines will follow each with a time in hh:mm format, where hh will mean the hour and mm will mean the minute. hh will be an integer within 0 to 23, and mm will be an integer within 0 to 59.

Output

Print "Yes" (without quotes) if it's a prime time, "No" (without quotes) if it's not.

Example

Input:
```
307:1104:1707:09
```

Output:
```
YesNoNo
```

Added by: Iqram Mahmud
Date: 2011-07-03
Time limit: 1s
Source limit:50000B
Languages: All
Resource: Own

SPOJ Problem Set (tutorial)

9113. Digibomb!

Problem code: NSUJ03C

Nayeem rushed to you with a sad, helpless look in his eyes. One part of his program needs to multiply two integers and find the number of digit it has. Now the problem is these numbers can be very big - as big as 18 digits. Everytime Nayeem tries to multiply them they overflow - since long long can only handle numbers less then 2^63-1, which is roughly 19 digits.
Nayeem believes you are super smart and you can help him this time. Can you write him that part of the program?

Input

First line will contain number of test cases t.
Each case will have two integers a and b. Both are less than 10^19.

Output

For each case output how many digits will be there in the product of a,b.

Example

Input:
```
4100 10010000 10000100000 1000001000000001804289383 1000000000846930886
```

Output:
```
591137
```

Added by: Iqram Mahmud
Date: 2011-07-03
Time limit: 1s
Source limit:50000B
Languages: All
Resource: Own

SPOJ Problem Set (tutorial)

9114. String Game

Problem code: NSUJ02D

Natasha is playing an innocent game since she is very very bored. She initially wrote a string that only contains ’a’ or ’b’. After in each turn, that she replaces all the ’b’ with ’ab’ and all the ’a’ with ’b’.
For example if she starts with "abba". Next time she’d write "bababb", by replacing ’a’s with ’b’ and ’b’s with ’ab’. But in that way, after a while the string gets really really big. Now she wonders if she does this work for n times, what would be the length of the string?

Input

First line contains number of test cases - t. After that t lines will follow, each will have an initial string and n. You can assume the given string won’t have more than 100 characters and n will be a positive integer not more than 50.

Output

For each case output the length of the string after n turns.

Example

```
Input:
3abba 1abba 2abbaabbabababaabba 50
Output:
610426530329384
```

Added by: Iqram Mahmud
Date: 2011-07-03
Time limit: 1s
Source limit:50000B
Languages: All
Resource: Own

SPOJ Problem Set (tutorial)

9115. Prime Kitties

Problem code: NSUJ02E

In NumboLand, all the houses have kitties. The house numbered n will have a kitty of type k, if and only if house k doesn't have more than 1 type of kitty and the number n is divisible by k. For example house 30 can only have kitties of type 2, 3 and 5. If you wonder why 1 is not there, note that house 1 is the kitty playground and you can find all type of kitties there.
House 3 will have only kitty of type 3. Because 3 divides 3 and it's not possible to have another type of kitty in that house maintaining the criterias. Same thing is true for house 5,11,19. Euclid proved that there are infinite number of houses who will have exactly one kitty. Everyone of NumboLand were excited when he did that. They threw a BIG party for that!
The owner of house n wants to visit other houses with all his kitties. He will not go to any house that has bigger number than his house. But there is a big problem, if he goes to any house that has the same type of kitties that he has, they'll start playing together and leave for kitty playground. And he doesn't want that to happen.
Can you tell the owner of house n, how many houses he can visit without having this incident?

Input

A single number t, number of test cases. After that there will be t lines, each containing one number n (2<=n<=1000000).

Output

For each case, output how many houses that owner can visit without losing any of his kitties.

Example

```
Input:
523417100
Output:
0111539
```

Added by: Iqram Mahmud
Date: 2011-07-03
Time limit: 1s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

9129. MAXNUM

Problem code: MAXNUM

Cho 2 sô nguyen du+o+ng N, P <= 30000. Tim sô M lo+'n nhât tho?a man P^M la u+o+'c cu?a N!

Input

Gôm 2 sô nguyen du+o+ng N va P

Output

Ghi ra duy nhât 1 ke^'t qua? cu?a bai toán. Test luôn đa?m ba?o có nghie^.m

Example

```
Input:
7 3

Output:
2
```

Added by: Minh^^
Date: 2011-07-06
Time limit: 0.029s-1s
Source limit:50000B
Languages: All except: CLOJ ERL F# GO PERL 6 PYTH 3.1.2 SCALA TCL

SPOJ Problem Set (tutorial)

9173. Repetitions

Problem code: REPTTS

A sequence of words over alphabet $['a',...,'z']$ is given. The length of longest word occuring as a coherent fragment in every word given is to be found.

Input

In the first line of the standard input there is an integer n , where $1 \le n \le 5$ is the number of words. In each of the next n lines there is one word formed from small letters of English alphabet $['a',...,'z']$. The length of each word is at least 1 , but not greater than $2\,000$.

Output

The text of standard output should consist of exactly one line containing a single integer equal to the length of the longest word occuring as the coherent fragment in every word given.

Example

Input:

```
3
abcb
bca
acbc
```

Output: 2

Added by: Krzysztof Lewko
Date: 2011-07-13
Time limit: 1s
Source limit:50000B
Languages: All
Resource: VII POI

SPOJ Problem Set (tutorial)

9339. Contains Prime

Problem code: AUCSE015

We will try investigating if a number contains a prime within it.

Input

The first line contains an integer T(1 <= T <= 100) which denotes the number of test cases.

Each test case contains a single integer N(100 <= N <= 1000000000).

Output

For each test case print "YES" if its possible to choose a subsequence of length 3 of the number which is a prime else print "NO" (without quotes).

Note: Consider the given number as a string of digits. So subsequences of this number refer to the numbers that can be formed by deleting zero or more digits from the original number and concatenating the left ones.

For eg,

Subsequence of 1234 having a length 3 are,

123

124

234

134

Example

```
Input:
2

141

1411856718567

Output:
NO

YES
```

```
For the second test case, we can choose the subsequences, 157, 167 etc which are prime
```

Added by: amit karmakar
Date: 2011-08-07
Time limit: 3s
Source limit:50000B
Languages: All

SPOJ Problem Set (classical)

9489. Johny Hates Math

Problem code: ANARC07J

Johnny is on probation! He has failed so many math courses and the Department has forced him to register in a remedial math course. He must pass the course or he'd be expelled from the University. In an attempt to impress his professor, Johnny is typing all his assignments on the computer. The latest assignment is rather simple, Johnny was given a set of problems to solve. Each problem had a list of one or more numbers that Johnny was supposed to add. Johnny has worked all night on the assignment, neatly typing his solution to each problem using a word processor as seen here:

4+12+3=19

As usual, Johnny woke up late, he hardly had the time to print the assignment and rush to class. Only in the classroom did he discover that, due to a printer driver problem, non of the plus signs were printed. The above line was printed as:

4123=19

Write a program to figure out where the pluses are supposed to be. All what Johnny remembers is that all the numbers were positive; None of the numbers, other than possibly the sum, had more than 5 digits; And none of the numbers had a zero as the left-most digit.

Input

The input consists of N (N <= 100) test cases. The number of them (N) is given on the first line of the input file. N lines follows each have an expression. No line will be longer than 256 characters.

Output

For each expression in the input, your program must print a line of*result. result*is the expression with the necessary plus signs in place. There are no spaces in*result*. If there are more than one possible solution, print a solution that requires the least number of plus signs. Knowing how bad Johnny is in arithmetic, it is possible that there is no solution, in which case your program should print ‘‘IMPOSSIBLE" as the*result*.

Example

Input:

```
3
4123=19
15442147612367219875=472
111=8
```

Output:

4+12+3=19
15+44+21+47+61+23+67+21+98+75=472
IMPOSSIBLE

Added by:	Amr Mesbah
Date:	2011-09-02
Time limit:	1s-30s
Source limit:	50000B
Languages:	All
Resource:	ANARC 2007

SPOJ Problem Set (tutorial)

9500. Bit counts

Problem code: BITCNTR

You are a detective working with Scotland Yard on a high-profile bank robbery case. The robber has left some trails in the form of numbers, which you believe will give you important clues about the robbery. Since you are genius, you know that the clues can obtained by converting the number in binary and counting the number of 1's.

Input

The first line has a number t followed by t lines. Each line contains a number < 512.

Output

Output t lines corresponding to each number, such that each of the line contains the clue obtained from the number.

Example

```
Input:
5231011

Output:
12103
```

Added by: Siddharth Kothari
Date: 2011-09-07
Time limit: 1s
Source limit:50000B
Languages: All
Resource: Own

SPOJ Problem Set (tutorial)

9501. Best balanced diet

Problem code: SETRR

Nutritionists have been struggling since a long time to determine what would constitute the best balance diet. You are famous for your fine programming skills, and hence as a test to prove your talent, you take up this challenge. You are given the nutrition value of all the possible items for producing the best balanced diet. Your task is to figure out which of these items to use and which to remove to produce the best balanced diet.

Input

The first line contains the number of itemsE. The next line contains E space separated integers, the actual nutrition value of each of the items. Value of E <= 10. Each nutrition value is in the range of [-2^31, 2^31).

Output

Output a single line in the following format "S N1 N2 .. NS", where S is the number of items in the best balanced diet, and N1, .., NS are the nutrition values of each of these items.

Example

```
Input:
4-4 0 4 1

Output:
2 4 1
```

Added by: Siddharth Kothari
Date: 2011-09-07
Time limit: 1s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

9514. Parsing URL

Problem code: TETRSIGM

In computing, a Uniform Resource Locator or Universal Resource Locator (URL) is a character string that specifies where a known resource is available on the Internet and the mechanism for retrieving it.

The syntax of a typical URL is:

scheme://domain:port/path?query_string#fragment_id

In this problem, the scheme, domain is required by all URL and other components are optional. That is, for example, the following are all correct urls:

```
http://dict.bing.com.cn/#%E5%B0%8F%E6%95%B0%E7%82%B9
http://www.mariowiki.com/Mushroom
https://mail.google.com/mail/?shva=1#inbox
http://en.wikipedia.org/wiki/Bowser_(character)
ftp://fs.fudan.edu.cn/
telnet://bbs.fudan.edu.cn/
http://mail.bashu.cn:8080/BsOnline/
```

Your task is to find the domain for all given URLs.

Input

There are multiple test cases in this problem. The first line of input contains a single integer denoting the number of test cases.

For each of test case, there is only one line contains a valid URL.

Output

For each test case, you should output the domain of the given URL.

Example

```
Input:
3
http://dict.bing.com.cn/#%E5%B0%8F%E6%95%B0%E7%82%B9
http://www.mariowiki.com/Mushroom
https://mail.google.com/mail/?shva=1#inbox

Output:
Case #1: dict.bing.com.cn
Case #2: www.mariowiki.com
Case #3: mail.google.com
```

Added by:	[Trichromatic] XilinX
Date:	2011-09-08
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	Fudan University Local Contest #3, by g201513

SPOJ Problem Set (classical)

9526. Department

Problem code: DEPART

The Department of Security has a new headquarters building. The building has several floors, and on each floor there are rooms numbered *xxyy* where *yy* stands for the room number and *xx* for the floor number, $0 < xx, yy < 10$. The building has 'pater-noster' elevator, i.e. elevator build up from several cabins running all around. From time to time the agents must visit the headquarters. During their visit they want to visit several rooms and in each room they want to stay for some time. Due to the security reasons, there can be only one agent in the same room at the same time, The same rule applies to the elevators. The visits are planned in the way ensuring they can be accomplished within one day. Each agent visits the headquarters at most once a day.

Each agent enters the building at the 1st floor, passes the reception and then starts to visit the rooms according to his/her list. Agents always visit the rooms by the increasing room numbers. The agents form a linear hierarchy according to which they have assigned their one letter personal codes. The agents with higher seniority have lexicographically smaller codes. No two agents have the same code.

If more then one agent want to enter a room, or an elevator, the agents have to form a queue. In each queue, they always stand according to their codes. The higher the seniority of the agent, the closer to the top of the queue he stands. Every 5 s (seconds) the first agent in the queue in front of the elevator enters the elevator. After visiting the last room in the headquarters each agent uses if necessary elevator to the first floor and exits the building.

The times necessary to move from a certain point in the headquarters to another are set as follows: Entering the building, i.e. passing the reception and reaching the elevator, or a room on the first floor takes 30 s. Exiting the building, i.e. stepping out of the elevator or a room on the first floor and passing the reception takes also 30 s. On the same floor, the transfer from the elevator to the room (or to the queue in front of the room), or from the room to the elevator (or to the queue in front of the elevator), or from one room to another (or to the queue in front of the room) takes 10 s. The transfer from one floor to the next floor above or below in an elevator takes 30 s. Write a program that determines time course of agent's visits in the headquarters.

Input

The input file contains the descriptions of $n \geq 0$ visits of different agents. The first line of the description of each visit consists of agent's one character code *C*, *C* = `A, ..., Z`, and the time when the agent enters the headquarters. The time is in the format HH:MM:SS (hours, minutes, seconds). The next lines (there will be at least one) contain the room number, and the length of time intended to stay in the room, time is in seconds. Each room is in a separate line. The list of rooms is sorted according to the increasing room number. The list of rooms ends by the line containing 0. The list of the descriptions of visits ends by the line containing the character dot.

Output

The output contains detailed records of each agent's visit in the headquarters. For each agent, there will be a block. Blocks are ordered in the order of increasing agent's codes. The first line of a block contains the code of agent. Next lines contain the starting and ending time (in format HH:MM:SS) and the descriptions of his/her activity. Time data will be separated by one blank character. Description will be separated from time by one blank character. Description will have a form`Entry`, `Exit`or Message. The Message can be one of the following:`Waiting in elevator queue`, `Waiting in front of room`*RoomNumber*,`Transfer from room`*RoomNumber*`to room`*RoomNumber*,`Transfer from elevator to room`*RoomNumber*,`transfer from`*RoomNumber*`to elevator`,`Stay in room`*RoomNumber*,`Stay in elevator`.
Print a blank line after each block.

Sample Input

```
A 10:00:00
0101 100
0110 50
0202 90
0205 50
0
B 10:01:00
0105 100
0201 5
0205 200
0
.
```

Sample Output

```
A
10:00:00 10:00:30 Entry
10:00:30 10:02:10 Stay in room 0101
10:02:10 10:02:20 Transfer from room 0101 to room 0110
10:02:20 10:03:10 Stay in room 0110
10:03:10 10:03:20 Transfer from room 0110 to elevator
10:03:20 10:03:50 Stay in elevator
10:03:50 10:04:00 Transfer from elevator to room 0202
10:04:00 10:05:30 Stay in room 0202
10:05:30 10:05:40 Transfer from room 0202 to room 0205
10:05:40 10:07:40 Waiting in front of room 0205
10:07:40 10:08:30 Stay in room 0205
10:08:30 10:08:40 Transfer from room 0205 to elevator
10:08:40 10:09:10 Stay in elevator
10:09:10 10:09:40 Exit

B
10:01:00 10:01:30 Entry
10:01:30 10:03:10 Stay in room 0105
10:03:10 10:03:20 Transfer from room 0105 to elevator
```

```
10:03:20 10:03:25 Waiting in elevator queue
10:03:25 10:03:55 Stay in elevator
10:03:55 10:04:05 Transfer from elevator to room 0201
10:04:05 10:04:10 Stay in room 0201
10:04:10 10:04:20 Transfer from room 0201 to room 0205
10:04:20 10:07:40 Stay in room 0205
10:07:40 10:07:50 Transfer from room 0205 to elevator
10:07:50 10:08:20 Stay in elevator
10:08:20 10:08:50 Exit
```

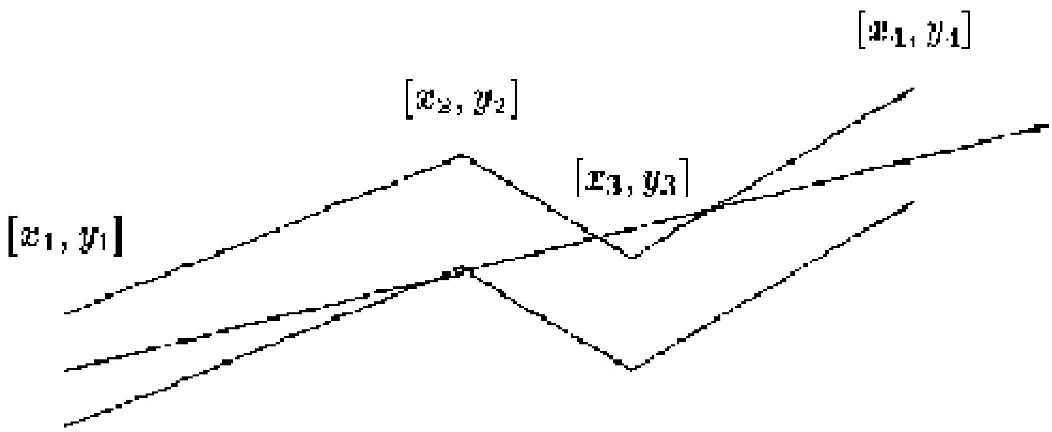
Added by: Andres Tellez
Date: 2011-09-12
Time limit: 1s
Source limit: 50000B
Languages: All

SPOJ Problem Set (classical)

9530. Pipe

Problem code: PIPEJ

The GX Light Pipeline Company started to prepare bent pipes for the new transgalactic light pipeline. During the design phase of the new pipe shape the company ran into the problem of determining how far the light can reach inside each component of the pipe. Note that the material which the pipe is made from is not transparent and not light reflecting.



Each pipe component consists of many straight pipes connected tightly together. For the programming purposes, the company developed the description of each component as a sequence of points $[x_1, y_1], [x_2, y_2], \ldots, [x_n, y_n]$, where $x_1 < x_2 < \ldots < x_n$ These are the upper points of the pipe contour. The bottom points of the pipe contour consist of points with *y*-coordinate decreased by 1. To each upper point $[x_i, y_i]$ there is a corresponding bottom point $[x_i, y_i - 1]$ (see picture above). The company wants to find, for each pipe component, the point with maximal *x*-coordinate that the light will reach. The light is emitted by a segment source with endpoints $[x_1, y_1 - 1]$ and $[x_1, y_1]$ (the endpoints are emitting light too). Assume that the light is not bent at the pipe bent points and the bent points do not stop the light beam.

Input

The input file contains several blocks each describing one pipe component. Each block starts with the number of bent points $2 \leq n \leq 20$ on separate line. Each of the next *n* lines contains a pair of real values x_i, y_i separated by space. The last block is denoted with $n = 0$.

Output

The output file contains lines corresponding to blocks in input file. To each block in the input file there is one line in the output file. Each such line contains either a real value, written with precision of two decimal places, or the message `Through all the pipe.`. The real value is the desired maximal x-coordinate of the point where the light can reach from the source for corresponding pipe component. If this value equals to x_n, then the message `Through all the pipe.` will appear in the output file.

Sample Input

```
4
0 1
2 2
4 1
6 4
6
0 1
2 -0.6
5 -4.45
7 -5.57
12 -10.8
17 -16.55
0
```

Sample Output

```
4.67
Through all the pipe.
```

Added by:	Andres Tellez
Date:	2011-09-12
Time limit:	3s
Source limit:	50000B
Languages:	All

SPOJ Problem Set (classical)

9531. Sticks

Problem code: STIJ

George took sticks of the same length and cut them randomly until all parts became at most 50 units long. Now he wants to return sticks to the original state, but he forgot how many sticks he had originally and how long they were originally. Please help him and design a program which computes the smallest possible original length of those sticks. All lengths expressed in units are integers greater than zero.

Input

The input file contains blocks of 2 lines. The first line contains the number of sticks parts after cutting. The second line contains the lengths of those parts separated by the space. The last line of the file contains zero.

Output

The output file contains the smallest possible length of original sticks, one per line.

Sample Input

```
9
5 2 1 5 2 1 5 2 1
4
1 2 3 4
0
```

Sample Output

```
6
5
```

Added by: Andres Tellez
Date: 2011-09-12
Time limit: 3s
Source limit:50000B
Languages: All

SPOJ Problem Set (tutorial)

9536. Calculating very big numbers very quickly

Problem code: MODPOW

Given numbers A, B, C, calculate (A to the power of B) mod C.

Input

The first line will contain an integer x, the number of test cases. x lines follow, each with three integers A, B, and C, the A, B, and C mentioned above.

Conditions:

A is no more than 1e5

B is no more than 1e18

C is no more than 1e7

Output

For each test case, print (A^B) mod C.

Example

```
Input:
11 1 2Output:
1
```

Added by: hua
Date: 2011-09-12
Time limit: 8s
Source limit:50000B
Languages: All
Resource: Alex Anderson

SPOJ Problem Set (tutorial)

9573. Spot the largest string

Problem code: SPOTIT

Rat Ronnie is very intelligent. Recently she got interested in the binary number system. Seeing this Rat Rocky decided to give her a problem to solve. If she solves it then she gets a big piece of cheese as a prize :).
A binary string of length N is a string that contains N characters. Each of these characters is either 0 or 1. Given a binary string S of length N and another input integer L, find a substring of length exactly L whose decimal value is largest amongst all substrings of length L in S. Print this largest value. (See notes and examples for further clarification)
Now Rat Ronnie is unable to think of anything else but cheese. As you are a brilliant programmer, she wants you to solve the problem. She promises to share the piece of cheese if you succeed.

Rat Ronnie is very intelligent. Recently she got interested in the binary number system. Seeing this Rat Rocky decided to give her a problem to solve. If she solves it then she gets a big piece of cheese as a prize :).

A binary string of length **N** is a string that contains N characters. Each of these characters is either 0 or 1. Given a binary string S of length **N** and another input integer **L**, find a substring of length **exactly** L whose decimal value is largest amongst all substrings of length L in S. Print this largest value. (See notes and examples for further clarification)

Now Rat Ronnie is unable to think of anything else but cheese. As you are a brilliant programmer, she wants you to solve the problem. She promises to share the piece of cheese if you succeed.

Notes

- A substring of a string S, is any contiguous sequence of characters in the string. For example, "cde" is a substring of "abcdef" but "ce" is not a substring of "abcdef".
- A value of a binary substring is the value after converting it to a decimal number. For example- Decimal value of "1101" = (2^0)*1 + (2^1)*0 + (2^2)*1 + (2^3)*1 = 13

Input

The first line is T, the number of test cases.

T test case follows. The first line of every test case contains two integers N and L. The second line of every test case contains a binary string of length N.

1<=T<=100

1<=N<=100

1<=L<=50

N>=L

Output

Output the maximum decimal value of the substring of length L. As the output may be large, use an appropriate data type.

Example

```
Input:
3
7 3
0110111
5 3
10110
4 4
1000

Output:
7
6
8
```

Explanation of Example

```
In the second test case, possible substrings of length 3 are "101" , "011", "110" . Out of these, "110" has the highest value in decimal, i.e, 6.
```

Added by:	Ishani Parekh
Date:	2011-09-22
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	Own

SPOJ Problem Set (tutorial)

9594. Count the strings

Problem code: NUMSTR

Alphabets made of cheese are available in the market these days. Rat Ronnie bought many alphabets from the market. After placing these alphabets one after another in a straight line (i.e a string of alphabets) Rat Ronnie went to sleep.
Rat Rocky who is always hungry, ate some of the cheese alphabets and escaped. Now Rat Ronnie is very furious. The string of alphabets that she had created was to be used in one of her science experiments. She doesnt remember the original string anymore. All she remembers is that each alphabet was a lower case english character. She wants you to find out the number of possible strings that she could have formed before Rat Rocky ate some of the characters.
(SeeA letter is a lower case english alphabet (i.e it can be any symbol from ’a’ to ’z’).
Letters made of cheese are available in the market these days. Rat Ronnie bought many letters from the market. After placing these letters one after another in a straight line (i.e a string of letters) Rat Ronnie went to sleep.
Rat Rocky who is always hungry, ate some of the cheese letters and escaped. Now Rat Ronnie is very furious. The string of letters that she had created was to be used in one of her science experiments. She doesnt remember the original string anymore. She wants you to find out the number of possible strings that she could have formed before Rat Rocky ate some of the characters.
(See examples and Notes for more clarification)A letter is a lower case english alphabet (i.e it can be any symbol from ’a’ to ’z’).
Letters made of cheese are available in the market these days. Rat Ronnie bought many letters from the market. After placing these letters one after another in a straight line (i.e a string of letters) Rat Ronnie went to sleep.
Rat Rocky who is always hungry, ate some of the cheese letters and escaped. Now Rat Ronnie is very furious. The string of letters that she had created was to be used in one of her science experiments. She doesnt remember the original string anymore. She wants you to find out the number of possible strings that she could have formed before Rat Rocky ate some of the characters.
(See examples and Notes for more clarification)

A letter is a lower case english alphabet (i.e it can be any symbol from ’a’ to ’z’).

Letters made of cheese are available in the market these days. Rat Ronnie bought many letters from the market. After placing these letters one after another in a straight line (i.e a string of letters) Rat Ronnie went to sleep.

Rat Rocky who is always hungry, ate some of the cheese letters and escaped. Now Rat Ronnie is very furious. The string of letters that she had created was to be used in one of her science experiments. She doesnt remember the original string anymore. She wants you to find out the number of possible strings that she could have formed before Rat Rocky ate some of the characters.

(See examples and Notes for more clarification)

Input

First line contains T , number of test cases. Next T lines will contain a string of characters which may have any number of spaces anywhere in the string (including the beginning). The last character of the input string will not be a space. Each input string terminates with a newline character.

A space in the input string denotes an unknown character (i.e a character that Rat Rocky ate)

T<=100

Each string will not exceed 100 characters.

Output

For each test case output the number of strings that Rat Ronnie could have formed on a new line. As the answer may be huge, output the answer modulo 10000007.

Notes

a modulo m : means the remainder after a is divided by m. It is also denoted by a%m.

If a,b are non negative integers, then the following hold

- (a+b)%m = ((a%m) + (b%m))%m
- (a*b)%m = ((a%m) * (b%m))%m

Example

Input:`4abc dega cz  d` **Output:**`12626676`

ExplanationFor the first test case, as there are no spaces in the string, it is the only possible string Rat Ronnie could have written.For the third test case "a c" the middle character can be any of the lower case 26 english alphabets.

Added by: Ishani Parekh
Date: 2011-09-27
Time limit: 1s
Source limit:50000B
Languages: All
Resource: Own

SPOJ Problem Set (tutorial)

9595. Fast addition

Problem code: MEGAUSS

Your friend Wannabe_Gauss thinks he is quite fast at mathematics. When his teacher asked him to find out whether the sum of all numbers in a given set of numbers was divisible by 2 or not, he answered it all quite fast.

Annoyed by this, his teacher gives him a big list of such sets of numbers, and he realizes he is after all a wannabe. He is relying on your programming prowess to get past this hurdle. Don't disappoint him!

Input

A number **t** on the first line, showing number of test cases.
Each test case begins with a number **n** on first line showing number of numbers in that test case. The next **n** lines contain one exactly one number each.

Output

T lines in output, one for each test case. "Y" if sum of that set is divisible by 2, else "N".

Example

Input:

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

Output:

```
N
Y
N
```

Constraints: `0 < t <= 50, 1 <= n <= 5000, each number is guaranteed to fit into a 32 bit integer.`

Added by:	Siddharth Kothari
Date:	2011-09-27
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	Chinmay Modi

SPOJ Problem Set (tutorial)

9598. Get me correct

Problem code: GETCORR

We already know the answer to *Life, the universe and everything*. It's the number 42. Now, we would like to know if a set of numbers satisfy this answer or not. We say the set satisfies if all the members of the set are divisible by this answer.

Input

There are going to be 6 members in the set to be tested. Each line specifies one member by an integer which fits into 32-bits.

Output

Print "Yes" if it satisfies, "No" otherwise.

Example

```
Input:
42428401261764

Output:
Yes
```

Added by:	Siddharth Kothari
Date:	2011-09-28
Time limit:	1s
Source limit:	50000B
Languages:	All
Resource:	own

SPOJ Problem Set (classical)

9654. Prime Count

Problem code: PCOUNT

How many primes are there between 1 and n where 1<=n<=10^8. Remember that 1 is not prime.

Input

The number n.

Output

Output the number of primes between 1 and n (inclusive).

Example

Input: 5 **Output:**3

Added by:	Paul Draper
Date:	2011-10-06
Time limit:	0.5s-10s
Source limit:	50000B
Languages:	All