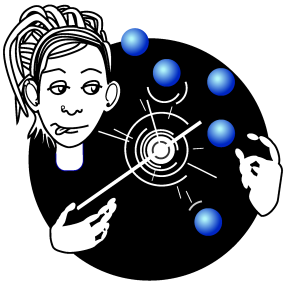
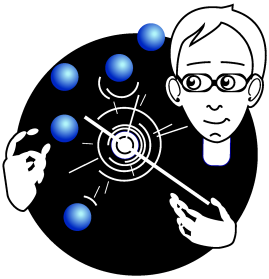
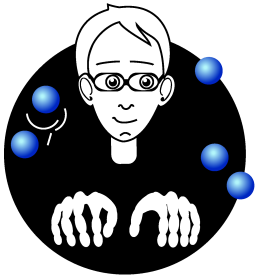
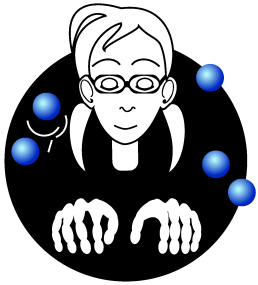


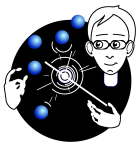


SOI 2010

The Swiss Olympiad in Informatics







SOI 2010

General information

What is the SOI?

The Swiss Olympiad in Informatics is the national qualification for the IOI, its international counterpart. The SOI aims to select and educate the four students that represent our country at the International Olympiad in Informatics.

What is the IOI?

The International Olympiad in Informatics is one of the scientific olympiads. It is the world championship in programming. Each year next to 300 participants from about 80 different countries compete against each other to win gold, silver or bronze medal.

Why to participate

- You learn a lot of interesting things about computer science.
- You meet a lot of people with similar or the same interests.
- You can win a free trip to a foreign country.
- You can participate in three camps that are not solely about programming.

What about the different tracks

For the first qualification round there are two tracks: a *practical track*, which focuses on the actual programming and a *theoretical track* which aims at the mathematical and computer science aspects. You may choose to follow either track to participate (or both if you like).

How can I get to the IOI?

The participants with the highest scores in the theoretical and practical track (distributed evenly) proceed to the second round. The best Round 1 participants will also be invited to a preparation camp in Davos – a five days long camp held in February, where the best participants are trained on IOI like tasks, take interesting lectures and have the opportunity to meet each other.

Round 2 will be a mixed round consisting of an online contest with practical tasks and an on-site theoretical test. The best participants of Round 2 are invited to the Final Round, which is composed of four programming contests on two weekends, held on-site in Zurich.

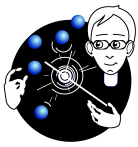
The best four students travel to *Canada* where the IOI 2010 takes place.

Rules

- You can participate if you have been born on 1. July 1990 or later and you are enrolled at a school of secondary edutation (high-school, vocational school, etc) during the period September to December 2009.
- To participate, you need to register at the web page of SOI: `http://www.soi.ch/` and set your role as 'participant'
- You must participate under your own name.
- We strongly encourage you to study from books and internet and exchange ideas with friends but do not copy any solutions (neither source code nor the description) from these sources.
- Solve the tasks on your own.

Information specific to the theoretical track

The tasks of theoretical track of Round 1 are presented in this document. You can find more information and hints about writing solutions of the theoretical track on our web page `http://www.soi.ch/contest/round1`.



How to submit your solution

After you have registered for the SOI 2010, you can submit your solution in four different ways:

1. By webpage: Upload your solution directly using the submit-forms which are linked on the webpage.

2. By email: theory@soi.ch
   Make sure you send only one mail with your final solutions for the tasks. You can send the solutions as PDF, Postscript, OpenOffice document, Word document or as a (reasonably sized) scanned image.

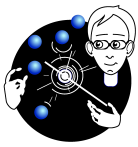
3. By mail: You can send a letter with all your solutions at
   Schweizer Informatikolympiade
   Theory A
   ETH Zürich
   CAB F 14
   Universitätstrasse 6
   8092 Zürich

4. By fax: You can send a fax with your solutions to +41 44 632 13 90.
   Make sure you write "SOI 2010" at the top of the Fax!

Whichever way you choose, please specify the name under which you have registered at the SOI webpage.

Until when to submit your solution

The deadline for the theoretical track of Round 1 is **30th of November 2009**.



Tasks of the Theoretical Track

1. Too Many Parentheses

The holiday is over and Mouse Stofl is again working hard on his math homeworks. He is almost finished by now, but he would like to check the correctness of his formulas once again. As a first step, he would like to check if the placement of the parentheses is correct. Unfortunately, his formulas contain lots of parentheses and checking their correctness is very time consuming. Write a program that helps Stofl with this tedious work!

Task: Your program should read a sequence of parentheses, i.e., a string that contain only characters `'('` and `')'`. Next, your program should check if the given sequence of parentheses is correct, i.e., if the sequence can be obtained by taking some correctly written formula and erasing all letters except parentheses.

Example:

Input:	**Output:**
(()())()	Sequence is correct.
Input:	**Output:**
)(	Sequence is not correct.

Explanation: In the first example, the sequence can be obtained e.g. from formula $((1+2)*(3+4))/(1+1)$. The second sequence is not correct, because it is not possible to use right parenthesis before the occurrence of the first left parenthesis.

2. The New Scarf

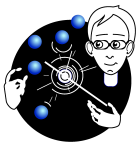
The winter is coming, hence Mouse Stofl's grandmother decided to make him a beautiful new scarf. She is very modern mouse and therefore she would like to knit him a peace of a modern art: She will use two different knitting patterns that we represent by 0 or 1. Each row of the scarf will contain N of these knitting patterns. Overall, there are 2^N possibilities how these 0s and 1s can be aligned in a row. Grandmother would like to use all of these 2^N different rows in the scarf, each exactly once, to make it very long. Moreover she would like to arrange these 2^N rows in such a way that two consecutive rows will differ in only one place.

The grandmother started to knit but after a few rows she became confused. While knitting a row, she forgot whether she already used the current row or not. Since keeping in mind all the rows used in the scarf is very hard for such an old mouse, she needs your help. Write her a program that will generate a sketch of the scarf so she will know how the scarf should look like.

Task: Write a program that for a given N generates the scarf sketch for Stofl's grandmother. The sketch consist of 0s and 1s that are aligned in overall 2^N rows that each have length N. Each row has to be different and the two consecutive rows have to differ in exactly one 0 or 1.

Input format: The only number in the input is the integer N representing the length of a row in the scarf and also determining the number of rows in the scarf.

Output format: Output 2^N rows each containing N 1s and 0s that represent the scarf's pattern. If there are more possible solutions, you can output any of them.



Example:

Input:	**Output:**
3	100 101 001 011 111 110 010 000

3. Strange Code

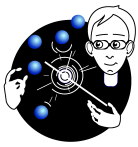
Stoffl's old good friend Mouse Gyver is really good at making cool gadgets. Right now, he is working on a new one. At first, he needs to calculate, how many centimeters of duct tape he will need. So, he wrote a small program to do the proper calculation. Unfortunately, Mouse Gyver is not very good at programming. Hence, even if he is absolutely sure that his code is correct, he is already waiting several hours for his program to finish.

Eventually, Mouse Gyver gives up and asks Stoffl for help. However, Gyver's code is so messy that even Mouse Stoffl's skills are not sufficient to figure out what exactly does the program calculate. Without your help, Gyver will never be able to finish his new cool gadget!

Task: You are given the Mouse Gyver's program below (there are C and Pascal versions that are equivalent). Figure out the output of the program and explain what does the program calculate.

```
#include <stdio.h>

#define N 314
#define M 3143149

int A[N], B[N], c;

void Prepare() {
```

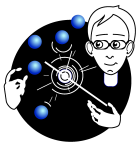
```
  int i,j,ok,g;
  g = N;
  ok = 0;
  while(g > 1 || ok == 0) {
    g = (int)((double)g/1.25);
    if (g < 1) g = 1;
    ok = 1;
    i = 0;
    while (i+g < N) {
      if (B[i] > B[i+g]) {
        j=B[i];
        B[i] = B[i+g];
        B[i+g] = j;
        ok = 0;
      }
      ++i;
    }
  }
}

void Check() {
  int i;
  for (i=0; i<N; i++) B[i] = A[i];
  Prepare();
  for (i=1; i<N; i++)
    if (B[i-1] == B[i]) return;
  c++;
  if (c==M) c = 0;
}

void Generate(int i) {
  if (i==N) Check();
  else {
    int j;
    for (j=0; j<N; j++) {
      A[i] = j;
      Generate(i+1);
    }
```



```
  }
}

int main() {
  c = 0;
  Generate(0);
  printf("%d\n", c);
  return 0;
}
```

```
program Gadget;

const N = 314;
      M = 3143149;

var A: array[1..N] of integer;
    B: array[1..N] of integer;
    c: integer;

procedure Prepare;
var i,j,g,ok:integer;
begin
  g:= N;
  ok:= 0;
  while (g > 1) or (ok=0) do
  begin
    g:= trunc(real(g)/1.25);
    if g < 1 then g:=1;
    ok:=1;
    i:=1;
    while i+g <= N do
    begin
      if V[i] > V[i+g] then
      begin
        j:=V[i];
        V[i]:=V[i+g];
        V[i+g]:=j;
        ok:=0;
```

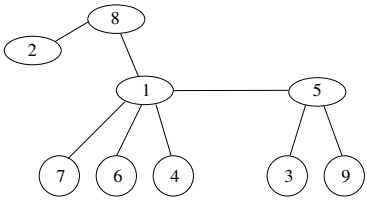
```
      end;
      i:=i+1;
    end;
  end;
end;

procedure Check;
var i:integer;
begin
  for i:=1 to N do B[i] := A[i];
  Prepare;
  for i:=2 to N do
    if B[i-1] = B[i] then exit;
  inc(c);
  if c = M then c := 0;
end;

procedure Generate(i:integer);
var j:integer;
begin
  if i=N+1 then Check()
  else
    for j:=1 to N do begin
      A[i] := j;
      Generate(i+1);
    end;
end;

begin
  c := 0;
  Generate(1);
  writeln(c);
end.
```



4. Chestnuts

In autumn, the fruits of most plants ripe and people harvest them for many reasons: Some fruits are processed into food, from some we obtain new seeds, and others are used to produce various materials, such as oil and cloth. The world of children is much simpler. They also gather the fruits, but their reason to do is pretty simple: Eat it if you can, play with it if you cannot.

This is also the case of mouse Stofl and his younger sister Lilly. While being in a forest, they have collected N horse chestnuts. Since these chestnuts can not be eaten, they use them for playing. First Stofl numbered these chestnuts from 1 to N to play a game of bingo. After he became bored with the game, he took wooden sticks and used all the chestnuts to build a figure in the shape of a cat with many legs:



Stofl's multileg-cat consists of the following chestnuts:

- four chestnuts represent the head, the neck, the chest, and the backside of the cat
- at least two chestnuts represent front paws
- at least two chestnuts represent back paws

Some pairs of chestnuts are connected by sticks: head and neck; neck and chest; chest and backside; each front paw and chest; and each back paw and backside of the cat.

Stofl has devised the following game: he used all of his N chestnuts (numbered from 1 to N) to build a valid multileg-cat. He then hide the cat away from Lilly. Lilly now has to identify all body parts of the cat. More precisely, for each number from 1 to N Lilly must find out which body part the chestnut with this number represents.

Stofl's rule of the game is that Lilly may only ask questions of the form "Are chestnuts x and y connected by a stick?", where x and y are the numbers

of two different chestnuts. Since Stofl knows the structure of the multileg-cat, he will correctly answer each such question.

Lilly does not like this game. She would prefer to have an algorithm that would play it in her place.

Task:

Help Lilly and design an algorithmic strategy how she should play. In other words, find an efficient algorithm that will read the value N and then ask some questions and make some computations in order to find the answer as quickly as possible.

In this task a detailed pseudocode [1] and the reasoning why the strategy works is sufficient. You do not have to write a program if you are sure that your pseudocode is clear. As a part of your solution include an estimate of the number of questions your program will ask in the worst possible case (in terms of N). Try to design an algorithm where the number of questions in the worst case is as small as possible.

When estimating the time complexity of your algorithm, you may assume that Stofl answers all the questions instantaneously and correctly.

Note that if you suggest a strategy for which the number of questions is quadratic in N (or worse), you can afford to ask Stofl about all pairs of chestnuts. However, such solution will not be sufficient to achieve the full score for this task.

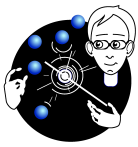
Example:

$N = 9$

Lilly: Are chestnuts 2 and 8 connected?	Stofl: Yes
Lilly: Are chestnuts 2 and 1 connected?	Stofl: No
Lilly: Are chestnuts 8 and 1 connected?	Stofl: Yes
Lilly: Are chestnuts 1 and 5 connected?	Stofl: Yes
Lilly: Are chestnuts 7 and 1 connected?	Stofl: Yes
Lilly: Are chestnuts 6 and 1 connected?	Stofl: Yes
Lilly: Are chestnuts 4 and 1 connected?	Stofl: Yes
Lilly: Are chestnuts 3 and 5 connected?	Stofl: Yes
Lilly: Are chestnuts 9 and 5 connected?	Stofl: Yes

Lilly: Head 2; neck 8; front part 1; back part 5; front legs 4, 6, 7; back legs 3, 9.

[1] In other words, give an exact high-level step by step description of your algorithm.



Clarification: For the purpose of grading, we do not make any difference between solutions with equal asymptotic number of questions. For example, solutions with $47N^2$ questions and $2N^2$ questions are for us the same.[2] Furthermore, the memory complexity of your solution will have no influence on the grading.

5. Planting Trees

When Mouse Stofl's great-grand uncle died, Stofl inherited a large parcel of land in Canada. The parcel has a shape of a square $N \times N$ meters long, and consists of $N \times N$ cells; each cell is a small square $1m \times 1m$.

As Stofl saw his parcel for the first time, he was somewhat disappointed. All his new land was completely empty, without a single tree. Since Stofl likes trees very much, he decided to plant some trees in his new parcel. Hence, he took N weeks of holiday and, in each week, he planned to plant one tree. However, he realized soon that he needs some help with keeping track of the positions of already planted trees. To make things even more complicated, some of the trees might die out due to the very harsh weather in Canada.

Task: Write a program that helps Stofl with keeping track of the positions of the trees. At first, you are given N – the size of the parcel and M – the number of Stofl's requests. There are two types of requests: Requests of first type inform your program about a position of a newly planted tree. If there is no living tree at this position, your program should output "Ok". Otherwise, your program should inform Stofl that there is already a living tree (so he can have a week off). Requests of the second type inform your program about a position of a tree that has died out. You may assume that the input is correct e.g., there is always a living tree in the position given by the request of the second type. There are exactly N requests of the first type.

Input format: At first your program should read N and M. Afterwards, M requests follows. Each request consists of 3 integers T – type of the request, X – the x-coordinate of the cell where the request occurs, and Y – the y-coordinate of the cell where the request occurs. It holds that $1 \leq X, Y \leq N$.

Output format: After reading every request of first type, your program should write "Ok" if there is no living tree in the given cell, or "Take a week off" otherwise. Do not output anything after reading a request of second type.

[2] If you want to know what exactly the word "asymptotic" means, read the introduction at `http://www.soi.ch/de/files/howto_en.pdf`.

Important notes: You may assume that your program has a lot of memory available (much more than N^2), but this memory is not initialized, i.e., its content is undefined. Try to design an algorithm with minimal possible time complexity, i.e., try to minimize the time required to process all requests. Note that initializing N^2 memory cells costs $O(N^2)$ time steps, and solutions working in $O(N^2)$ time will not receive full score.

Example:

Input:	**Output:**
4 6	Ok
1 1 1	Ok
1 3 3	Take a week off
1 1 1	Ok
2 1 1	
2 3 3	
1 1 1	

Explanation: At first, Stofl plants trees at $[1, 1]$ and $[3, 3]$. In the third week, there is a living tree on $[1, 1]$, so Stofl has a week off. Afterwards, both trees die, and Stofl plants another tree at $[1, 1]$.