Associação para a Promoção e Desenvolvimento da Sociedade da Informação

Olimpíadas Internacionais de Informática (IOI'2008)

Relatório de Actividades

Outubro 2008

Patrocínio Principal

Patrocínio

ÍNDICE

1. Introdução

As Olimpíadas Internacionais de Informática, IOI – International Olympiad in Informatics – http://www.ioinformatics.org/index.shtml são uma das seis olimpíadas de ciência, destinadas estudantes do ensino secundário de todo o mundo. As outras são as Olimpíadas da Matemática, da Física, da Química, da Biologia e da Astronomia.

O objectivo principal das Olimpíadas Internacionais de Informática é estimular o interesse dos jovens pela informática e pelas tecnologias da informação. Os vencedores das IOI em cada ano são estudantes excepcionais e pertencem ao grupo dos melhores jovens cientistas mundiais no domínio da Informática.

Este ano, a Olimpíada Internacional de Informática realizou-se no Cairo, Egipto, http://www.ioi2008.org/, entre 16 e 23 de Agosto, e Portugal foi um dos países participantes, entre mais de 80, com uma delegação formada por quatro estudantes e dois professores.

Portugal tem estado presente regularmente nas IOI desde 1992. A APDSI continua a conduzir a participação portuguesa. Portugal foi o país organizador da edição de 1998, que teve lugar em Setúbal e que trouxe até ao nosso país, no ano da Expo em Lisboa, mais de 300 estudantes de todo o mundo e seus acompanhantes.

A equipa portuguesa para o concurso no Cairo foi formada pelos estudantes António Silva da Escola Secundária Padre Benjamim Salgado, em Vila Nova de Famalicão, Pedro Abreu da Escola Secundária Jaime Moniz, no Funchal, Ricardo Martins da Escola S/3 D. Manuel I, em Beja e Tiago Andrade do Colégio Internato dos Carvalhos, em Vila Nova de Gaia. Estes quatro concorrentes foram seleccionados no final de um estágio de preparação que decorreu entre 28 de Julho e 1 de Agosto nas instalações do Departamento de Ciência de Computadores da Faculdade de Ciências da Universidade do Porto e no qual participaram os oito primeiros classificados das Olimpíadas Nacionais de Informática (ONI) deste ano.

Integraram a delegação portuguesa os professores Pedro Guerreiro e Pedro Ribeiro, da Universidade do Algarve e da Universidade do Porto, respectivamente. Estes dois professores foram os responsáveis técnicos e científicos pelas ONI e por todo o processo de selecção e treino dos concorrentes para as IOI.

2. Conclusões

Nos anos anteriores, os quatro melhores classificados nas Olimpíadas Nacionais de Informática eram seleccionados automaticamente para constituir a equipa portuguesa concorrente às Olimpíadas Internacionais. Desde 2005, decidimos proceder de maneira diferente, convidando para um estágio de formação os oito melhores e escolhendo a equipa só no final do estágio, por meio de um concurso realizado em moldes semelhantes aos das Olimpíadas Internacionais.

O objectivo desta medida era evitar que os concorrentes, seleccionados com muita antecedência, descurassem a sua preparação, e incrementar o empenho de todos no estágio. No final do estágio, os quatro escolhidos, não coincidiram com os quatro mais bem classificados nas Olimpíadas Nacionais.

O estágio realizou-se nas instalações do Departamento de Ciência de Computadores da Faculdade de Ciências da Universidade do Porto, de 28 de Julho a 1 de Agosto, isto é, duas semanas antes da prova internacional. Foi coordenado pelo Prof. Pedro Guerreiro e pelo Dr. Pedro Ribeiro, e contou com a colaboração de alguns concorrentes de anos anteriores – Filipe Brandão, André Santos e André Pinto – presentemente alunos da Universidade do Porto e ainda de Miguel Oliveira, também estudante da Universidade do Porto.

Durante o estágio, os alunos concorrentes aprendem as principais técnicas de programação necessárias para resolver os problemas que costumam sair neste tipo de concursos. Verificámos, após a prova no Egipto, que o conteúdo do estágio estava muito bem adequado ao que realmente saiu, ainda que, infelizmente, os alunos não tivessem sido capazes de pôr em prática os ensinamentos recebidos.

Os quatro alunos seleccionados – António Silva da Escola Secundária Padre Benjamim Salgado, em Vila Nova de Famalicão, Pedro Abreu da Escola Secundária Jaime Moniz, no Funchal, Ricardo Martins da Escola S/3 D. Manuel I, em Beja e Tiago Andrade do Colégio Internato dos Carvalhos, em Vila Nova de Gaia – viajaram para o Cairo, acompanhados pelo Prof. Pedro Guerreiro, *team leade*r, e pelo Dr. Pedro Ribeiro, *deputy leader*.

Nas Olimpíadas Internacionais de Informática, o concurso é constituído por duas provas, cada uma de cinco horas, em cada uma das quais são apresentados três problemas de programação que se encontram em anexo.

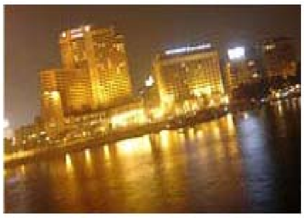
Apesar de bem preparados, os nossos concorrentes fraquejaram, e não resolveram devidamente problemas que normalmente estariam ao seu alcance. Por isso, a classificação final foi modesta, muito abaixo das nossas legítimas expectativas.

De acordo com as regras da Olimpíadas Internacionais, os concorrentes na primeira metade da classificação recebem uma medalha. Há medalhas de ouro, prata e bronze, distribuídas na proporção 1:2:3. Em anexo está a lista dos 140 medalhados, entre os quais não se encontra nenhum dos concorrentes portugueses. A classificação dos concorrentes não medalhados não é publicada, apenas é divulgada a pontuação de todos eles.

O sítio *web* das Olimpíadas Internacionais de Informática, 2008, cujo endereço é http://www.ioi2008.org/, contém mais informações sobre o evento, cuja página de entrada incluímos em anexo.

ANEXOS

Anexo I - Sítio na *web* (http://www.ioi2008.org/)



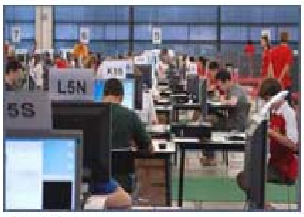
Egypt

Please check the Visa page...

Experience eternal Egypt.

Check the weather, currency & other practical information before you visit.

Fascinated by Pyramids & ancient civilizations? Read on to discover a country steeped in the past.



IOI 2008

Registration and Accommodation fees are now available...

View the I.C. meeting image gallery...

What is IOI?

Where will this year's IOI be held?

The International Olympiad in Informatics (IOI) is one of the most recognized computer science competitions in the world. The 20th International Olympiad in Informatics will be held in Egypt, August 2008, at Mubarak city for education located in the new district 6 october of Cairo metropolitan. Historically, Egypt is probably the world's oldest civilization having emerged from the Nile Valley to be a land bustling with life, beauty and excitement.

15% discount from EgyptAir

This is the logo for this year IOI 2008 in Egypt. Behind the choice of the logo a story. The shape of the logo is one of the famous Pharonic shapes, ...

Read more ++

HIS EXCELLENCY, PRESIDENT OF EGYPT, THE HONORARY PATRON

MOHAMED HOSNY MUBARAK

This event is organized under his honorable care. >>

IOI 2008 Calendar

October 2008

Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Weather

Cairo, Egypt

Temp:	29°C
Wind Chill:	29°C
Humidity:	37%

Anexo II - Problemas das duas Provas das IOI

TYPE PRINTER

You need to print ***N*** words on a movable type printer. Movable type printers are those old printers that require you to place small metal pieces (each containing a letter) in order to form words. A piece of paper is then pressed against them to print the word. The printer you have allows you to do any of the following operations:

- Add a letter to the end of the word currently in the printer.
- Remove the last letter from the end of the word currently in the printer. You are only allowed to do this if there is at least one letter currently in the printer.
- Print the word currently in the printer.

Initially, the printer is empty; it contains no metal pieces with letters. At the end of printing, you are allowed to leave some letters in the printer. Also, you are allowed to print the words in any order you like.

As every operation requires time, you want to minimize the total number of operations.

TASK

You must write a program that, given the ***N*** words you want to print, finds the minimum number of operations needed to print all the words in any order, and outputs one such sequence of operations.

CONSTRAINTS

1 <= ***N*** <= 25,000 The number of words you need to print.

INPUT

Your program must read from the standard input the following data:

- Line 1 contains the integer ***N***, the number of words you need to print.
- Each of the next ***N*** lines contains a word. Each word consists of lower case letters (‘a’ – ‘z’) only and has length between 1 and 20, inclusive.
  All words will be distinct.

OUTPUT

Your program must write to the standard output the following data:

- Line 1 must contain an integer ***M*** that represents the minimum number of operations required to print the ***N*** words.
- Each of the next ***M*** lines must contain one character each. These characters describe the sequence of operations done. Each operation must be described as follows:
  - Adding a letter is represented by the letter itself in lowercase
  - Removing the last letter is represented by the character ‘-‘ (minus, ASCII code 45)
  - Printing the current word is represented by the character ‘P’ (uppercase letter P)

GRADING

For a number of tests, worth a total of 40 points, N will not exceed 18.

DETAILED FEEDBACK

During the contest, your submissions for this task will be evaluated on some of the official test data, showing you a summary of the results.

EXAMPLE

Sample input	Sample output
3 print the poem	20 t h e P - - - p o e m P - - - r i n t P

FISH

It was told by Scheherazade that far away, in the middle of the desert, there is a lake. Originally this lake had ***F*** fish in it. ***K*** different kinds of gemstones were chosen among the most valuable on Earth, and to each of the ***F*** fish exactly one gem was given for it to swallow. Note, that since ***K*** might be less than ***F***, two or more fish might swallow gems of the same kind.

As time went by, some fish ate some of the other fish. One fish can eat another if and only if it is at least twice as long (fish ***A*** can eat fish ***B*** if and only if $L_A >= 2 * L_B$). There is no rule as to when a fish decides to eat. One fish might decide to eat several smaller fish one after another, while some fish may decide not to eat any fish, even if they can. When a fish eats a smaller one, its length doesn't change, but the gems in the stomach of the smaller fish end up undamaged in the stomach of the larger fish.

Scheherazade has said that if you are able to find the lake, you will be allowed to take out one fish and keep all the gems in its stomach for yourself. You are willing to try your luck, but before you head out on the long journey, you want to know how many different combinations of gems you could obtain by catching a single fish.

TASK

Write a program that given the length of each fish and the kind of gemstone originally swallowed by each fish, finds **the number of different combinations of gems that can end up in the stomach of any fish, <u>modulo some given integer *M*</u>.** A combination is defined only by the number of gems from each of the ***K*** kinds. There is no notion of order between gems, and any two gems of the same kind are indistinguishable.

CONSTRAINTS

1 <= ***F*** <= 500,000	The original number of fish in the lake.
1 <= ***K*** <= ***F***	The number of different gemstone kinds.
2 <=***M*** <= 30,000	
1 <= L_X <= 1,000,000,000	The length of fish *X.*

INPUT

Your program must read from the standard input the following data:

- Line 1 contains the integer ***F***, the original number of fish in the lake.
- Line 2 contains the integer ***K***, the number of kinds of gemstones.

The kinds of gemstones are represented by integers 1 to ***K***, inclusive.

- Line 3 contains the integer ***M***.
- Each of the following ***F*** lines describes one fish using 2 integers separated by a single space: the length of the fish followed by the kind of gemstone originally swallowed by that fish.

NOTE: For all test cases used for evaluation, it is guaranteed that there is at least one gemstone from each of the ***K*** kinds.

OUTPUT

Your program must write to the standard output a single line containing one integer between 0 and ***M-1*** (inclusive): the number of different possible combinations of gemstones **modulo *M***.
Note that for solving the task, the value of ***M*** has no importance other than simplifying computations.

GRADING

For a number of tests, worth a total of 70 points, ***K*** will not exceed 7,000.
Also, for some of these tests, worth a total of 25 points, ***K*** will not exceed 20.

DETAILED FEEDBACK

During the contest, your submissions for this task will be evaluated on some of the official test data showing you a summary of the results.

EXAMPLE

Sample Input	Sample Output
5 3 7 2 2 5 1 8 3 4 1 2 3	4

There are 11 possible combinations so you should output 11 modulo 7 which is 4.

The possible combinations are: [1] [1,2] [1,2,3] [1,2,3,3] [1,3] [1,3,3] [2] [2,3] [2,3,3] [3] and [3,3].
(For each combination, we list the gems it contains. For example, [2,3,3] is a combination that consists of one gem of kind 2, and two gems of kind 3.)
These combinations can be achieved in the following ways:

- [1]: It is possible that you catch the second (or the fourth) fish before it eats any other fish.
- [1,2]: If the second fish eats the first fish, then it would have a gemstone of kind 1 (the one it originally swallowed) and a gemstone of kind 2 (from the stomach of the first fish).
- [1,2,3]: One possible way of reaching this combination: the fourth fish eats the first fish, and then the third fish eats the fourth fish. If you now catch the third fish, it will have one gemstone of each kind in its stomach.
- [1,2,3,3]: Fourth eats first, third eats fourth, third eats fifth, you catch the third one.
- [1,3]: Third eats fourth, you catch it.
- [1,3,3]: Third eats fifth, third eats fourth, you catch it.
- [2]: You catch the first fish.
- [2,3]: Third eats first, you catch it.
- [2,3,3]: Third eats first, third eats fifth, you catch it.
- [3]: You catch the third fish.
- [3,3]: Third eats fifth, you catch it.

ISLANDS

You are visiting a park which has ***N*** islands. From each island ***i***, exactly one bridge was constructed. The length of that bridge is denoted by L_i. The total number of bridges in the park is ***N***. Although each bridge was built from one island to another, now every bridge can be traversed in both directions. Also, for each pair of islands, there is a unique ferry that travels back and forth between them.

Since you like walking better than riding ferries, you want to maximize the sum of the lengths of the bridges you cross, subject to the constraints below.

- You can start your visit at an island of your choice.
- You may not visit any island more than once.
- At any time you may move from your current island ***S*** to another island ***D*** that you have **not** visited before. You can go from ***S*** to ***D*** either by:
  - Walking: Only possible if there is a bridge between the two islands. With this option the length of the bridge is added to the total distance you have walked, or
  - Ferry: You can choose this option only if ***D*** is not reachable from ***S*** using any combination of bridges and/or previously used ferries. (When checking whether it is reachable or not, you consider all paths, including paths passing through islands that you have already visited.)

Note that you do not have to visit all the islands, and it may be impossible to cross all the bridges.

TASK

Write a program that, given the ***N*** bridges along with their lengths, computes the maximum distance you can walk over the bridges obeying the rules described above.

CONSTRAINTS

2 <= ***N*** <= 1,000,000	The number of islands in the park.
$1 <= L_i <= 100,000,000$	The length of bridge ***i***.

INPUT

Your program must read from the standard input the following data:

- Line 1 contains the integer ***N***, the number of islands in the park. Islands are numbered from 1 to ***N***, inclusive.
- Each of the next ***N*** lines describes a bridge. The i^{th} of these lines describes the bridge constructed from island ***i*** using two integers separated by a single space. The first integer represents the island at the other endpoint of the bridge, the second integer represents the length L_i of the bridge. You may assume that for each bridge, its endpoints are always on two different islands.

OUTPUT

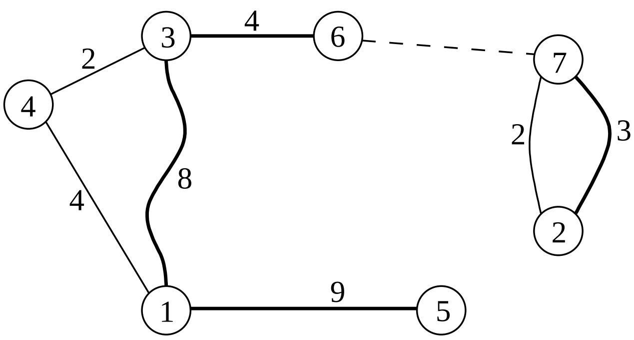
Your program must write to the standard output a single line containing one integer, the maximum possible walking distance.

NOTE 1: For some of the test cases the answer will not fit in a 32-bit integer, you might need int64 in Pascal or long long in C/C++ to score full points on this problem.
NOTE 2: When running Pascal programs in the contest environment, it is significantly slower to read in 64-bit data types than 32-bit data types from standard input even when the values being read in fit in 32 bits. We recommend that you read the input data into 32-bit data types.

GRADING
For some cases worth 40 points, ***N*** will not exceed 4,000.

EXAMPLE

Sample input	Sample output
7 3 8 7 2 4 2 1 4 1 9 3 4 2 3	24



The ***N=7*** bridges in the sample are (1-3), (2-7), (3-4), (4-1), (5-1), (6-3) and (7-2). Note that there are two different bridges connecting islands 2 and 7.
One way that you can achieve maximum walking distance follows:

- Start on island 5.
- Walk the bridge of length 9 to reach island 1.
- Walk the bridge of length 8 to reach island 3.
- Walk the bridge of length 4 to reach island 6.
- Take the ferry from island 6 to island 7.
- Walk the bridge of length 3 to reach island 2.

By the end you are on island 2 and your total walking distance is 9+8+4+3 = 24.
The only island that was not visited is island 4. Note that at the end of the trip described above you can not visit this island any more. More precisely:

- You are not able to visit it by walking, because there is no bridge connecting island 2 (where you currently stand) and island 4.
- You are not able to visit it using a ferry, because island 4 is reachable from island 2, where you currently stand. A way to reach it: use the bridge (2-7), then use a ferry you already used to get from island 7 to island 6, then the bridge (6-3), and finally the bridge (3-4).

LINEAR GARDEN

Ramesses II has just returned victorious from battle. To commemorate his victory, he has decided to build a majestic garden. The garden will contain a long line of plants that will run all the way from his palace at Luxor to the temple of Karnak. It will consist only of lotus plants and papyrus plants, since they symbolize Upper and Lower Egypt respectively.

The garden must contain exactly ***N*** plants. Also, it must be balanced: in any contiguous section of the garden, the numbers of lotus and papyrus plants must not differ by more than 2.

A garden can be represented as a string of letters 'L' (lotus) and 'P' (papyrus). For example, for ***N=5*** there are 14 possible balanced gardens. In alphabetical order, these are: LLPLP, LLPPL, LPLLP, LPLPL, LPLPP, LPPLL, LPPLP, PLLPL, PLLPP, PLPLL, PLPLP, PLPPL, PPLLP, and PPLPL.

The possible balanced gardens of a certain length can be ordered alphabetically, and then numbered starting from 1. For example, for ***N=5***, garden number 12 is the garden PLPPL.

TASK
Write a program that, given the number of plants ***N*** and a string that represents a balanced garden, calculates the number assigned to this garden **modulo some given integer *M*.**
Note that for solving the task, the value of ***M*** has no importance other than simplifying computations.

CONSTRAINTS
1 <= ***N*** <= 1,000,000
7 <= ***M*** <= 10,000,000

GRADING
In inputs worth a total of 40 points, ***N*** will not exceed 40.

INPUT
Your program must read from the standard input the following data:

- Line 1 contains the integer ***N***, the number of plants in the garden.
- Line 2 contains the integer ***M***.
- Line 3 contains a string of ***N*** characters 'L' (lotus) or 'P' (papyrus) that represents a balanced garden.

OUTPUT
Your program must write to the standard output a single line containing one integer between 0 and ***M-1*** (inclusive), the number assigned to the garden described in the input, **modulo *M***.

EXAMPLE

Sample input 1	Sample output 1
5 7 PLPPL	5

The actual number assigned to PLPPL is **12**. So, the output is **12** modulo **7,** which is **5**.

Sample input 2	Sample output 2
12 10000 LPLLPLPPLPLL	39

TELEPORTERS

You are participating in a competition that involves crossing Egypt from west to east along a straight line segment. Initially you are located at the westmost point of the segment. It is a rule of the competition that you must always move along the segment, and always eastward.

There are N teleporters on the segment. A teleporter has two endpoints. Whenever you reach one of the endpoints, the teleporter immediately teleports you to the other endpoint. (Note that, depending on which endpoint of the teleporter you reach, teleportation can transport you either eastward or westward of your current position.) After being teleported, you must continue to move eastward along the segment; you can never avoid a teleporter endpoint that is on your way. There will never be two teleporter endpoints at the same position. Endpoints will be strictly between the start and the end of the segment.

Every time you get teleported, you earn 1 point. The objective of the competition is to earn as many points as possible. In order to maximize the points you earn, you are allowed to add up to M new teleporters to the segment before you start your journey. You also earn points for using the new teleporters.

You can set the endpoints of the new teleporters wherever you want (even at non-integer coordinates) as long as they do not occupy a position already occupied by another endpoint. That is, the positions of the endpoints of all teleporters must be unique. Also, endpoints of new teleporters must lie strictly between the start and the end of the segment.

Note that it is guaranteed that no matter how you add the teleporters, you can always reach the end of the segment.

TASK

Write a program that, given the position of the endpoints of the N teleporters, and the number M of new teleporters that you can add, computes the maximum number of points you can earn.

CONSTRAINTS

$1 <= N <= 1,000,000$	The number of teleporters initially on the segment.
$1 <= M <= 1,000,000$	The maximum number of new teleporters you can add.
$1 <= W_X < E_X <= 2,000,000$	The distances from the beginning of the segment to the western and eastern endpoints of teleporter X.

INPUT

Your program must read from the standard input the following data:

- Line 1 contains the integer N, the number of teleporters initially on the segment.
- Line 2 contains the integer M, the maximum number of new teleporters that you can add.
- Each of the next N lines describes one teleporter. The i^{th} of these lines describes the i^{th} teleporter. Each line consists of 2 integers: W_i and E_i separated by a space. They represent respectively the

distances from the beginning of the segment to the western and eastern endpoints of the teleporter.

No two endpoints of the given teleporters share the same position. The segment that you will be travelling on starts at position 0 and ends at position 2,000,001.

OUTPUT

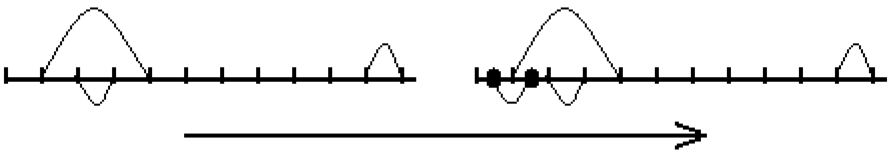
Your program must write to the standard output a single line containing one integer, the maximum number of points you can earn.

GRADING

In test data worth 30 points, ***N*** <= 500 and ***M*** <= 500.

EXAMPLE

Sample input 1	Sample output 1
3 1 10 11 1 4 2 3	6



The first figure shows a segment with the three original teleporters. The second figure shows the same segment after adding a new teleporter with endpoints at 0.5 and at 1.5.

After adding the new teleporter as shown in the figure, your travel would be the following:

- You start at position 0, moving eastward.
- You reach the endpoint at position 0.5 and get teleported to position 1.5 (you earn 1 point).
- You continue to move east and reach endpoint at position 2; you get teleported to position 3 (you have 2 points).
- You reach endpoint at position 4, and get teleported to 1 (you have 3 points).
- You reach endpoint at 1.5, and get teleported to 0.5 (you have 4 points).
- You reach endpoint at 1, and get teleported to 4 (you have 5 points).
- You reach endpoint at 10, and get teleported to 11 (you have 6 points).
- You continue until you reach the end of the segment finishing with a total score of 6 points.

Sample input 2	Sample output 2
3 3 5 7 6 10 1999999 2000000	12

PYRAMID BASE

You have been asked to find the largest affordable location for constructing a new pyramid. In order to help you decide, you have been provided with a survey of the available land which has been conveniently divided into an M by N grid of square cells. The base of the pyramid must be a square with sides parallel to those of the grid.

The survey has identified a set of P possibly overlapping obstacles, which are described as rectangles in the grid with sides parallel to those of the grid. In order to build the pyramid, all the cells covered by its base must be cleared of any obstacles. Removing the i^{th} obstacle has a cost C_i. Whenever an obstacle is removed, it must be removed completely, that is, you cannot remove only part of an obstacle. Also, please note that removing an obstacle does not affect any other obstacles that overlap it.

TASK

Write a program that, given the dimensions M and N of the survey, the description of the P obstacles, the cost of removing each of the obstacles, and the budget B you have, finds the maximum possible side length of the base of the pyramid such that the total cost of removing obstacles does not exceed B.

CONSTRAINTS AND GRADING

Your program will be graded on three disjoint sets of tests. For all of them, the following constraints apply:

$1 <= M, N <= 1{,}000{,}000$	The dimensions of the grid.
$1 <= C_i <= 7{,}000$	The cost of removing the i^{th} obstacle.
$1 <= X_{i1} <= X_{i2} <= M$	X coordinates of the leftmost and the rightmost cells of the i^{th} obstacle.
$1 <= Y_{i1} <= Y_{i2} <= N$	Y coordinates of the bottommost and the topmost cells of the i^{th} obstacle.

<u>In the first set of tests worth 35 points:</u>

$B = 0$	The budget you have. (You cannot remove any obstacles.)
$1 <= P <= 1{,}000$	The number of obstacles in the grid.

<u>In the second set of tests worth 35 points:</u>

$0 < B <= 2{,}000{,}000{,}000$	The budget you have.
$1 <= P <= 30{,}000$	The number of obstacles in the grid.

<u>In the third set of tests worth 30 points:</u>

$B = 0$	The budget you have. (You cannot remove any obstacles.)
$1 <= P <= 400{,}000$	The number of obstacles in the grid.

INPUT

Your program must read from the standard input the following data:

- Line 1 contains two integers separated by a single space that represent M and N respectively.
- Line 2 contains the integer B, the maximum cost you can afford (i.e., your budget).
- Line 3 contains the integer P, the number of obstacles found in the survey.

- Each of the next ***P*** lines describes an obstacle. The i^{th} of these lines describes the i^{th} obstacle. Each line consists of 5 integers: X_{i1}, Y_{i1}, X_{i2}, Y_{i2}, and C_i separated by single spaces. They represent respectively the coordinates of the bottommost leftmost cell of the obstacle, the coordinates of the topmost rightmost cell of the obstacle, and the cost of removing the obstacle. The bottommost leftmost cell on the grid has coordinates (1, 1) and the topmost rightmost cell has coordinates (***M***, ***N***).

OUTPUT

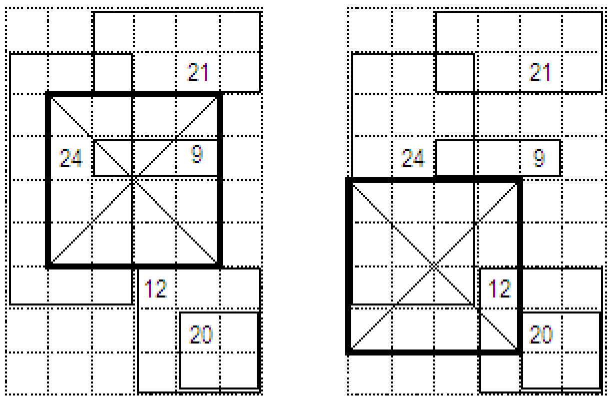
Your program must write to the standard output a single line containing one integer, the maximum possible side length of the base of the pyramid that can be prepared. If it is not possible to build any pyramid, your program should output the number 0.

DETAILED FEEDBACK

During the contest, your submissions for this task will be evaluated on some of the official test data showing you a summary of the results.

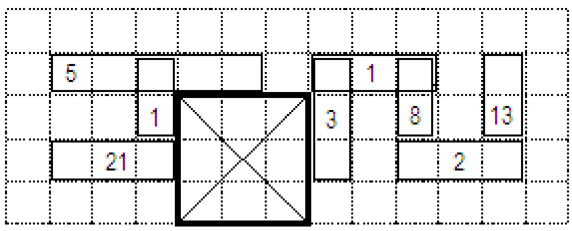
EXAMPLE

Sample input 1	Sample output 1
6 9 42 5 4 1 6 3 12 3 6 5 6 9 1 3 3 8 24 3 8 6 9 21 5 1 6 2 20	4



The figure shows two possible locations for the pyramid's base, both having a side of length 4.

Sample input 2	Sample output 2
13 5 0 8 8 4 10 4 1 4 3 4 4 1 10 2 12 2 2 8 2 8 4 3 2 4 6 4 5 10 3 10 4 8 12 3 12 4 13 2 2 4 2 21	3



The figure shows the only possible location for the pyramid's base having a side of length 3.

Anexo III - Classificações das IOI

Results

IOI 2008 Results

Rank	Contestant	Country	typ	fsh	isl	tel	lin	pbs	Score
1	Huacheng Yu	China	100	100	88	100	100	70	558
2	Panupong Pasupat	Thailand	100	67	62	100	100	45	474
3	Dong Zhou	China	100		100	100	100	70	470
3	Marcin Koscielnicki	Poland	100		100	100	100	70	470
5	Goran Zuzic	Croatia	100	6	100	100	100	60	466
6	Cosmin Gheorghe	Romania	100	65	100	6	100	70	441
7	Henadzi Karatkevich	Belarus	100	45	22	100	100	70	437
8	David Benjamin	United States	100	57	62	100	100	10	429
9	Jack Murray	Australia	100	0	86	100	100	35	421
10	Neal Wu	United States	100	10	100	100	100	10	420
11	Julian Fischer	Germany	100		57	100	100	60	417
12	Chan min Kim	Republic of Korea	100	10	60	100	72	70	412
13	Danqi Chen	China	100	67	74	100	34	35	410
14	Marcin Andrychowicz	Poland	100	10	29	100	100	70	409
15	Sergey Rogulenko	Russian Federation	100	5	100	68	100	35	408
16	Han jay Yang	Chinese Taipei	100	0	62	100	100	35	397
17	Andras Eisenberger	Hungary	100	70	10	100	100	10	390
18	Jaroslaw Blasiok	Poland	100	6	23	100	100	60	389
19	Hanson Wang	Canada	100	10	67	100	100	10	387
20	Tana Wattanawaroon	Thailand	100		67	100	100	15	382
21	Vladislav Epifanov	Russian Federation	100	2	29	100	100	35	366
22	Irvan Jahja	Indonesia	100	0	26	100	100	35	361
23	Kazuhiro Hosaka	Japan	100	10	35	100	100	15	360
24	Che yang Wu	Chinese Taipei	100	2	100	49	90	15	356
25	Makoto Soejima	Japan	100	0	48	69	100	35	352
26	Maciej Klimek	Poland	100	10	100		100	35	345
27	Jae sung Park	Republic of Korea	100	0	100	10	93	35	338
28	Pavels Cupikovs	Latvia	100	0	33	74	100		307
29	Zhomart Sadykov	Kazakhstan	40	7	77	69	100	10	303
29	Jacob Steinhardt	United States	100		68	0	100	35	303
31	Sun il Kwon	Republic of Korea	100	0	23	7	100	70	300
32	Wei quan Lim	Singapore	100		62		100	30	292
33	Vladimir Boza	Slovakia	100	4	26	30	100	25	285
34	Paul Baltescu	Romania	100	10	40	1	100	30	281
35	Balazs Szalkai	Hungary	100	7	68	68	36	0	279
36	Johannes Josi	Switzerland	86		62	28	100		276
37	Brian Hamrick	United States	100	2	24	100	34	15	275
38	Yury Pisarchyk	Belarus	100	0	0	69	100	5	274
38	Ronald Chan	New Zealand	100	0	16	58	100		274
40	Momchil Tomov	Bulgaria	100		53	5	100	15	273
40	Linyun Yu	China	100	7	8	23	100	35	273
42	Ivo Sluganovic	Croatia	100	10	62		100		272
42	Akim Kumok	Russian Federation	100	10	68	19	40	35	272
44	Abel Nieto Rodriguez	Cuba	100		36		100	35	271
45	Abhabongse Janthong	Thailand	100	0	40	68	61		269
46	Wiktor Jakubiuk	United Kingdom	48	0	50	70	100	0	268
47	Miroslav Klimos	Czech Republic	100	0	62	5	100	0	267
48	Jelle Hooff	Netherlands	100		5	24	100	35	264
49	Miroslaw Michalski	United Kingdom	100	5	50	8	100	0	263
50	Cheuk ting Li	Hong Kong , China	20		26	100	100	15	261
51	Raman Udavichenka	Belarus	100	0	7	7	100	45	259
52	Ji hoon Ryoo	Republic of Korea	100	0	29	4	100	25	258
53	Massimo Cairo	Italy	86	0		49	100	20	255
53	Erik Massop	Netherlands	100		27	48	60	20	255
55	Ruslan Simonenko	Ukraine	100	0	53		100		253
56	Bakhytzhan Baizhikenov	Kazakhstan	100	0	26		100	25	251

57	Gediminas Liktaras	Lithuania	100			68	80		248
57	Enrique Lira Vargas	Mexico	100	0	33	0	100	15	248
59	Nhan Nguyen	Vietnam	100	5	42		100		247
60	Harry Slatyer	Australia	100	0	3	28	100	15	246
60	Alexander Kaluzhin	Russian Federation	30	10	48	18	100	40	246
62	Gunnar Peng	Sweden	100	10	40	6	58	30	244
63	Jarrah Lacko	Australia	100	10	62	11	55	5	243
63	Peter Ondruska	Slovakia	100	10	13	0	80	40	243
65	Miklos Danka	Hungary	100	0	11	15	100	15	241
66	Vladimir Serbinenko	Switzerland	30	2	2	100	100	5	239
66	Son Nguyen	Vietnam	100	0	8	6	90	35	239
68	You cheng Syu	Chinese Taipei	100	0	17	16	100	0	233
69	Rumen Hristov	Bulgaria	90		24	6	100	10	230
69	Frederick Manners	United Kingdom	10	67	0	68	85		230
71	Roman Smrz	Czech Republic	100		21		93	15	229
72	Patrik Fimml	Austria	100		23		100	5	228
73	Louis Jachiet	France	100	11	16		100	0	227
74	Sergiy Mogylnyy	Ukraine	100		26		100	0	226
75	Arif Yildiz	Turkey	100	0	14	11	100		225
76	Doeke Wolf	Netherlands	100			0	85	35	220
77	Giovanni Mascellani	Italy	40	0	0	78	100	0	218
77	Visit Pataranutaporn	Thailand	100	0	3	0	100	15	218
79	Robin Cheng	Canada	100	3	21	0	90		214
79	Ioannis Chatzimichos	Greece	72	7	62		58	15	214
79	Man hin Hon	Hong Kong, China	100		3	100	11	0	214
82	Ming fung Tai	Hong Kong, China	100		13		100	0	213
82	Reinardus Pradhitya	Indonesia	100	10	22	0	71	10	213
84	Adas Burksaitis	Lithuania	100		0		100	10	210
84	Wei zhong Lim	Singapore	100		0		100	10	210
86	Paolo Comaschi	Italy	90	0	21	0	76	20	207
87	Aliaksei Ropan	Belarus	30		33	5	100	35	203
88	Srivatsan Balakrishnan	India	100	0			72	30	202
88	Vytautas Gruslys	Lithuania	72	0	0		100	30	202
88	Jia han Chiam	Singapore	100	2			100	0	202
91	Eiichi Matsumoto	Japan	78		0	68	55		201
92	Eike Mueller	Germany	100				100	0	200
92	Gabor Danner	Hungary	100		0		100		200
92	Peter Fulla	Slovakia	100	0			100	0	200
95	Bogdan Tantsyura	Ukraine	92			5	100		197
96	Quang Vu	Vietnam	100				90	5	195
97	Shuo fu Chen	Chinese Taipei	100	0	7	0	85	0	192
98	Martin Fixman	Argentina	100	5	40	5	40	0	190
98	Stefan Avramov	Bulgaria	100				80	10	190
100	Petrus Risan	Indonesia	100	7	40	0	22	15	184
101	Maja Kabiljo	Serbia	100	0	14	7	58		179
102	Martin Marinov	Bulgaria	100	0	40		22	15	177
103	Ahmed Ezzat	Egypt	100		40		35		175
104	Ludwig Schmidt	Germany	72	0			100	0	172
105	Mikko Sysikaski	Finland	56	0		7	98	5	166
105	Motoki Takigiku	Japan	100		30		26	10	166
107	Marcelo Povoa	Brazil	40	4	10	1	100	10	165
108	Martun Karapetyan	Armenia	100	2	0	1	40	20	163
108	Aleksi Hartikainen	Finland	20	0	18		90	35	163
110	Anders Christiansen	Denmark	100	0	9		53	0	162
111	Jan Bercic	Slovenia	100		21		40		161
112	Gabriel Dalalio	Brazil	30	0	29		100	0	159
113	Muhammad Moolla	South Africa	56	0			100	0	156
114	Hayk Saribekyan	Armenia	100	0	19		35		154
114	Olegs Osmjans	Latvia	100	0	29	5		20	154
116	Nicolas Ponieman	Argentina	40		18	0	95		153
116	Listiarso Wastuargo	Indonesia	100	0	26	7	5	15	153
118	Adrian satja Kurdija	Croatia	30	7	0		100	15	152
118	Bruno Rahle	Croatia	100	0	11	1	40		152
118	Christophe Kamphaus	Luxembourg	100		20		22	10	152
118	Bogdan Tataroiu	Romania	100	0	21		26	5	152
122	Simon Lindholm	Sweden	100	10			40	0	150
122	Dat Nguyen	Vietnam	20		0	0	95	35	150
124	Aaron Voelker	Canada	100	0	8		40	0	148
124	Zurab Kutsia	Georgia	100	0	6	5	22	15	148
124	Mehmet Saka	Turkey	20	0	23		90	15	148
127	Angel Pina	Spain	62	0	23		62	0	147
127	Levan Varamashvili	Georgia	100	10	9	6	22	0	147

Results

IOI 2008 Results

Rank	Contestant	Country	typ	fsh	isl	tel	lin	pbs	Score
1	Huacheng Yu	China	100	100	88	100	100	70	558
2	Panupong Pasupat	Thailand	100	67	62	100	100	45	474
3	Dong Zhou	China	100		100	100	100	70	470
3	Marcin Koscielnicki	Poland	100		100	100	100	70	470
5	Goran Zuzic	Croatia	100	6	100	100	100	60	466
6	Cosmin Gheorghe	Romania	100	65	100	6	100	70	441
7	Henadzi Karatkevich	Belarus	100	45	22	100	100	70	437
8	David Benjamin	United States	100	57	62	100	100	10	429
9	Jack Murray	Australia	100	0	86	100	100	35	421
10	Neal Wu	United States	100	10	100	100	100	10	420
11	Julian Fischer	Germany	100		57	100	100	60	417
12	Chan min Kim	Republic of Korea	100	10	60	100	72	70	412
13	Danqi Chen	China	100	67	74	100	34	35	410
14	Marcin Andrychowicz	Poland	100	10	29	100	100	70	409
15	Sergey Rogulenko	Russian Federation	100	5	100	68	100	35	408
16	Han jay Yang	Chinese Taipei	100	0	62	100	100	35	397
17	Andras Eisenberger	Hungary	100	70	10	100	100	10	390
18	Jaroslaw Blasiok	Poland	100	6	23	100	100	60	389
19	Hanson Wang	Canada	100	10	67	100	100	10	387
20	Tana Wattanawaroon	Thailand	100		67	100	100	15	382
21	Vladislav Epifanov	Russian Federation	100	2	29	100	100	35	366
22	Irvan Jahja	Indonesia	100	0	26	100	100	35	361
23	Kazuhiro Hosaka	Japan	100	10	35	100	100	15	360
24	Che yang Wu	Chinese Taipei	100	2	100	49	90	15	356
25	Makoto Soejima	Japan	100	0	48	69	100	35	352
26	Maciej Klimek	Poland	100	10	100		100	35	345
27	Jae sung Park	Republic of Korea	100	0	100	10	93	35	338
28	Pavels Cupikovs	Latvia	100	0	33	74	100		307
29	Zhomart Sadykov	Kazakhstan	40	7	77	69	100	10	303
29	Jacob Steinhardt	United States	100		68	0	100	35	303
31	Sun il Kwon	Republic of Korea	100	0	23	7	100	70	300
32	Wei quan Lim	Singapore	100		62		100	30	292
33	Vladimir Boza	Slovakia	100	4	26	30	100	25	285
34	Paul Baltescu	Romania	100	10	40	1	100	30	281
35	Balazs Szalkai	Hungary	100	7	68	68	36	0	279
36	Johannes Josi	Switzerland	86		62	28	100		276
37	Brian Hamrick	United States	100	2	24	100	34	15	275
38	Yury Pisarchyk	Belarus	100	0	0	69	100	5	274
38	Ronald Chan	New Zealand	100	0	16	58	100		274
40	Momchil Tomov	Bulgaria	100		53	5	100	15	273
40	Linyun Yu	China	100	7	8	23	100	35	273
42	Ivo Sluganovic	Croatia	100	10	62		100		272
42	Akim Kumok	Russian Federation	100	10	68	19	40	35	272
44	Abel Nieto Rodriguez	Cuba	100		36		100	35	271
45	Abhabongse Janthong	Thailand	100	0	40	68	61		269
46	Wiktor Jakubiuk	United Kingdom	48	0	50	70	100	0	268
47	Miroslav Klimos	Czech Republic	100	0	62	5	100	0	267
48	Jelle Hooff	Netherlands	100		5	24	100	35	264
49	Miroslaw Michalski	United Kingdom	100	5	50	8	100	0	263
50	Cheuk ting Li	Hong Kong , China	20		26	100	100	15	261
51	Raman Udavichenka	Belarus	100	0	7	7	100	45	259
52	Ji hoon Ryoo	Republic of Korea	100	0	29	4	100	25	258
53	Massimo Cairo	Italy	86	0		49	100	20	255
53	Erik Massop	Netherlands	100		27	48	60	20	255
55	Ruslan Simonenko	Ukraine	100	0	53		100		253
56	Bakhytzhan Baizhikenov	Kazakhstan	100	0	26		100	25	251

57	Gediminas Liktaras	Lithuania	100			68	80		248
57	Enrique Lira Vargas	Mexico	100	0	33	0	100	15	248
59	Nhan Nguyen	Vietnam	100	5	42		100		247
60	Harry Slatyer	Australia	100	0	3	28	100	15	246
60	Alexander Kaluzhin	Russian Federation	30	10	48	18	100	40	246
62	Gunnar Peng	Sweden	100	10	40	6	58	30	244
63	Jarrah Lacko	Australia	100	10	62	11	55	5	243
63	Peter Ondruska	Slovakia	100	10	13	0	80	40	243
65	Miklos Danka	Hungary	100	0	11	15	100	15	241
66	Vladimir Serbinenko	Switzerland	30	2	2	100	100	5	239
66	Son Nguyen	Vietnam	100	0	8	6	90	35	239
68	You cheng Syu	Chinese Taipei	100	0	17	16	100	0	233
69	Rumen Hristov	Bulgaria	90		24	6	100	10	230
69	Frederick Manners	United Kingdom	10	67	0	68	85		230
71	Roman Smrz	Czech Republic	100		21		93	15	229
72	Patrik Fimml	Austria	100		23		100	5	228
73	Louis Jachiet	France	100	11	16		100	0	227
74	Sergiy Mogylnyy	Ukraine	100		26		100	0	226
75	Arif Yildiz	Turkey	100	0	14	11	100		225
76	Doeke Wolf	Netherlands	100			0	85	35	220
77	Giovanni Mascellani	Italy	40	0	0	78	100	0	218
77	Visit Pataranutaporn	Thailand	100	0	3	0	100	15	218
79	Robin Cheng	Canada	100	3	21	0	90		214
79	Ioannis Chatzimichos	Greece	72	7	62		58	15	214
79	Man hin Hon	Hong Kong, China	100		3	100	11	0	214
82	Ming fung Tai	Hong Kong, China	100		13		100	0	213
82	Reinardus Pradhitya	Indonesia	100	10	22	0	71	10	213
84	Adas Burksaitis	Lithuania	100		0		100	10	210
84	Wei zhong Lim	Singapore	100		0		100	10	210
86	Paolo Comaschi	Italy	90	0	21	0	76	20	207
87	Aliaksei Ropan	Belarus	30		33	5	100	35	203
88	Srivatsan Balakrishnan	India	100	0			72	30	202
88	Vytautas Gruslys	Lithuania	72	0	0		100	30	202
88	Jia han Chiam	Singapore	100	2			100	0	202
91	Eiichi Matsumoto	Japan	78		0	68	55		201
92	Eike Mueller	Germany	100				100	0	200
92	Gabor Danner	Hungary	100		0		100		200
92	Peter Fulla	Slovakia	100	0			100	0	200
95	Bogdan Tantsyura	Ukraine	92			5	100		197
96	Quang Vu	Vietnam	100				90	5	195
97	Shuo fu Chen	Chinese Taipei	100	0	7	0	85	0	192
98	Martin Fixman	Argentina	100	5	40	5	40	0	190
98	Stefan Avramov	Bulgaria	100				80	10	190
100	Petrus Risan	Indonesia	100	7	40	0	22	15	184
101	Maja Kabiljo	Serbia	100	0	14	7	58		179
102	Martin Marinov	Bulgaria	100	0	40		22	15	177
103	Ahmed Ezzat	Egypt	100		40		35		175
104	Ludwig Schmidt	Germany	72	0			100	0	172
105	Mikko Sysikaski	Finland	56	0		7	98	5	166
105	Motoki Takigiku	Japan	100		30		26	10	166
107	Marcelo Povoa	Brazil	40	4	10	1	100	10	165
108	Martun Karapetyan	Armenia	100	2	0	1	40	20	163
108	Aleksi Hartikainen	Finland	20	0	18		90	35	163
110	Anders Christiansen	Denmark	100	0	9		53	0	162
111	Jan Bercic	Slovenia	100		21		40		161
112	Gabriel Dalalio	Brazil	30	0	29		100	0	159
113	Muhammad Moolla	South Africa	56	0			100	0	156
114	Hayk Saribekyan	Armenia	100	0	19		35		154
114	Olegs Osmjans	Latvia	100	0	29	5		20	154
116	Nicolas Ponieman	Argentina	40		18	0	95		153
116	Listiarso Wastuargo	Indonesia	100	0	26	7	5	15	153
118	Adrian satja Kurdija	Croatia	30	7	0		100	15	152
118	Bruno Rahle	Croatia	100	0	11	1	40		152
118	Christophe Kamphaus	Luxembourg	100		20		22	10	152
118	Bogdan Tataroiu	Romania	100	0	21		26	5	152
122	Simon Lindholm	Sweden	100	10			40	0	150
122	Dat Nguyen	Vietnam	20		0	0	95	35	150
124	Aaron Voelker	Canada	100	0	8		40	0	148
124	Zurab Kutsia	Georgia	100	0	6	5	22	15	148
124	Mehmet Saka	Turkey	20	0	23		90	15	148
127	Angel Pina	Spain	62	0	23		62	0	147
127	Levan Varamashvili	Georgia	100	10	9	6	22	0	147

127	Eduards Kalinicenko	Latvia	20			7	100	20	147
127	David Venhoek	Netherlands	100		12		35	0	147
127	Ugljesa Stojanovic	Serbia	36	0	11	0	100		147
132	Pedro da silva	Brazil	100		32			10	142
132	Zhan xiong Chin	Singapore	100	0	18	7	2	15	142
134	Rodrigo Santiago Nieves	Mexico	100			5	35	0	140
135	Rafael Mantilla	Colombia	100		0		39		139
136	Ricardo Pereira	Brazil	100				2	35	137
137	Mariam Kobiashvili	Georgia	100			5	31		136
138	Samuel Hapak	Slovakia	100	0	29		0		129
139	Nadeem Moidu	India	30		0		98		128
140	Xi Chen	Australia	100	0	19	0	8	0	127
140	Pradeep Mathias	India	86		0	1	40		127
142			72	0	0		51		123
142			100				8	15	123
144			56	0	29		35	0	120
144			30	0	33	7	50	0	120
146			64	0	21	5	27	0	117
147			10	10	0	5	85	5	115
148			0		14		100	0	114
149			0			6	100	5	111
149			100			11			111
151			100					10	110
151			48	0	21	1	40		110
151			20	2	40	7	26	15	110
151			10		0		100		110
155			74	0			35		109
156			40	0	17	5	46	0	108
156			0	0	8		100	0	108
158			100	0			7	0	107
158			40	10	2	5	40	10	107
160			100				6		106
160			56	0			35	15	106
162			72		6		26		104
163			100	0	0		2		102
163			40	0	62		0		102
165			40		21		40		101
166			100	0			0		100
166			100	0					100
168			30		21		30	15	96
169			72	0			22		94
169			54	0	40				94
171			40		7		40	5	92
171			20		40		22	10	92
173			20	0	25		31	10	86
174			30	0		28	20	5	83
175			40	0	40	0	2	0	82
176			30	2	7		22	15	76
176			30		11		35		76
176			30		11		35	0	76
179			38	0			35		73
180			30	7		0	35		72
181			10		21	7	33	0	71
181			20		11	5	35	0	71
183			20		13		22	15	70
184			10			0	58		68
184			20		32		16	0	68
186			30	0		15	21	0	66
186			66						66
188			30		10		22		62
189			10	0			51	0	61
190			20	0			40		60
190			20	0	0		40		60
192			20		6	5	26	0	57
193			46		0		10	0	56
194			10	0	14		31		55
194				0			40	15	55
196			20	2			32		54
197			10			10	33		53
197			30		18	0	5	0	53
197			20	0	20	13	0	0	53
200			10	10	0	0	22	10	52

127	Eduards Kalinicenko	Latvia	20			7	100	20	147
127	David Venhoek	Netherlands	100		12		35	0	147
127	Ugljesa Stojanovic	Serbia	36	0	11	0	100		147
132	Pedro da silva	Brazil	100		32			10	142
132	Zhan xiong Chin	Singapore	100	0	18	7	2	15	142
134	Rodrigo Santiago Nieves	Mexico	100			5	35	0	140
135	Rafael Mantilla	Colombia	100		0		39		139
136	Ricardo Pereira	Brazil	100				2	35	137
137	Mariam Kobiashvili	Georgia	100			5	31		136
138	Samuel Hapak	Slovakia	100	0	29		0		129
139	Nadeem Moidu	India	30		0		98		128
140	Xi Chen	Australia	100	0	19	0	8	0	127
140	Pradeep Mathias	India	86		0	1	40		127
142			72	0	0		51		123
142			100				8	15	123
144			56	0	29		35	0	120
144			30	0	33	7	50	0	120
146			64	0	21	5	27	0	117
147			10	10	0	5	85	5	115
148			0		14		100	0	114
149			0			6	100	5	111
149			100			11			111
151			100					10	110
151			48	0	21	1	40		110
151			20	2	40	7	26	15	110
151			10		0		100		110
155			74	0			35		109
156			40	0	17	5	46	0	108
156			0	0	8		100	0	108
158			100	0			7	0	107
158			40	10	2	5	40	10	107
160			100				6		106
160			56	0			35	15	106
162			72		6		26		104
163			100	0	0		2		102
163			40	0	62		0		102
165			40		21		40		101
166			100	0			0		100
166			100	0					100
168			30		21		30	15	96
169			72	0			22		94
169			54	0	40				94
171			40		7		40	5	92
171			20		40		22	10	92
173			20	0	25		31	10	86
174			30	0		28	20	5	83
175			40	0	40	0	2	0	82
176			30	2	7		22	15	76
176			30		11		35		76
176			30		11		35	0	76
179			38	0			35		73
180			30	7		0	35		72
181			10		21	7	33	0	71
181			20		11	5	35	0	71
183			20		13		22	15	70
184			10			0	58		68
184			20		32		16	0	68
186			30	0		15	21	0	66
186			66						66
188			30		10		22		62
189			10	0			51	0	61
190			20	0			40		60
190			20	0	0		40		60
192			20		6	5	26	0	57
193			46		0		10	0	56
194			10	0	14		31		55
194				0			40	15	55
196			20	2			32		54
197			10			10	33		53
197			30		18	0	5	0	53
197			20	0	20	13	0	0	53
200			10	10	0	0	22	10	52

201			10	0		5	35		50
202			20		14		0	15	49
203				0			40	5	45
204			30	0		10	2		42
204			0				22	20	42
206			20		21		0		41
206			20	0	21	0			41
208			0				40		40
208				0			40		40
208			0		0		40	0	40
208							35	5	40
212			10		3		24		37
212			20				2	15	37
214			10		0	16		10	36
214					32		4		36
216			0				35		35
216			0				35		35
216			10		3		22		35
219				0			33	0	33
219			0	0			13	20	33
221			10				22		32
221			10				22		32
223			0				26	5	31
224			10		5			15	30
225			20				8		28
226			20	0		5	2	0	27
227			0	0			26		26
228			10				15	0	25
229			10	5			8		23
230			0	0			20	0	20
230				0			0	20	20
230			20	0			0		20
233			0	0			0	15	15
233					0	0	0	15	15
235			0	0		0	2	10	12
235			10				2		12
237			10				0	0	10
237			10		0		0		10
237			10	0		0	0		10
237			0					10	10
237			10				0		10
237			10	0			0		10
237			10				0		10
244			0				8		8
245			0			0	2	5	7
246			0	0	0	4	2	0	6
246			0	0	0	0	6	0	6
246					0	6	0		6
249				0			0	5	5
249				0		5	0	0	5
249					0	5			5
249				0		5			5
253			0				2		2
253			0	0	0	0	2	0	2
253							2	0	2
253				0			2		2
253			0	0		0	2		2
258				0			0	0	0
258						0			0
258							0		0
258						0	0		0
258				0					0
258			0				0		0
258							0		0
258							0		0
258			0						0
258							0		0
258				0				0	0
258				0		0		0	0
258			0				0	0	0
258			0		0	0	0	0	0
258			0						0

258			**0**	**0**		**0**	**0**		**0**
258						**0**	**0**	**0**	**0**
258						**0**	**0**	**0**	**0**
258							**0**		**0**
258								**0**	**0**
258						**0**		**0**	**0**
258								**0**	**0**
258							**0**		**0**
258									**0**
258									**0**
258									**0**

Tree