Associação para a Promoção e Desenvolvimento da Sociedade da Informação

Olimpíadas Internacionais de Informática (IOI'2007)

Relatório de Actividades

Outubro 2007

apdsi
associação para a promoção e desenvolvimento da Sociedade da Informação

FCT FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE NOVA DE LISBOA

FACULDADE DE CIÊNCIAS FC U.PORTO

Patrocínio Principal

Caixa Geral de Depositos

Patrocínio

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

ÍNDICE

1. Introdução

As Olimpíadas Internacionais de Informática, IOI – International Olympiad in Informatics – http://olympiads.win.tue.nl/ioi, 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 Zagreb, na Croácia, http://www.hsin.hr/ioi2007/, entre 15 e 22 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, desde o ano passado, continua a conduziu 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 na Croácia foi formada pelos estudantes Filipe Brandão, Miguel Araújo, Nuno Lourenço e Tiago Andrade. Estes quatro concorrentes foram seleccionados no final de um estágio de preparação que decorreu entre 30 de Julho e 3 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. Dos quatro seleccionados, três, o Miguel Araújo, Nuno Lourenço e Tiago Andrade, são do Colégio Internato dos Carvalhos, de Vila Nova de Gaia, o Filipe Brandão é da Escola Secundária de Arouca.

Integraram a delegação portuguesa os professores Pedro Guerreiro e Pedro Ribeiro, da Universidade Nova de Lisboa 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.

3. 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 30 de Julho a 3 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 do senhor Pedro Pereira, aluno da licenciatura em Ciências da Computação da Universidade do Porto, antigo concorrente às Olimpíadas Internacionais.

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 na Croácia, 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 – Filipe Brandão, Miguel Araújo, Nuno Lourenço e Tiago Andrade – viajaram para a Croácia, 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 142 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, 2007, cujo endereço é http://www.hsin.hr/ioi2007/, contém mais informações sobre o evento, cuja página de entrada incluímos em anexo.

ANEXOS

Anexo I - Sítio na *web* (http://www.hsin.hr/ioi2007/)

- IOI
  - Introduction
  - History
  - Regulations
- IOI 2007
  - News
  - Organizers
  - Visas
  - Registration
  - Accommodation
  - Fees
  - Schedule
  - Venue
  - IC & ISC
  - March meeting
  - IOI Conference
  - Newsletters
  - Contact
- Competition
  - Rules
  - Tasks
  - Results
  - Online contest
- Location
  - About Croatia
  - About Zagreb
  - Gallery

Dear friends!

We are honoured to host the 19th International Olympiad in Informatics in Croatia, and we are inviting everyone to attend. We will do our best to make IOI 2007 a memorable occasion and hope you will meet old friends and make new ones.
Teams from all IOI countries have been invited to come to Croatia, and we will also try to get some new countries to join.

IOI - What is it?

The **International Olympiad in Informatics (IOI)** is one of the most prominent computer science competitions in the world. The Competition Tasks are of an algorithm nature, however the contestants have to show basic skills as problem analysis, design of the algorithm needed, data structures, as well as programming and testing of their solutions. The winners of the IOI belong, no doubt, to the best young computer scientists of the world.

The IOI is one of eight International Science Olympiads.
The other seven are: Mathematics IMO (since 1959), Physics IPhO (since 1967), Chemistry IChO (since 1968), Biology IBO (Since 1990), Astronomy IAO (since 1996), Geography IGEO (since 1996), and The International Linguistic Olympiad ILO (since 2003). All of them are open to High School students, younger than 20 years of age.

Each country participates at the IOI with a National Delegation, consisting of four Contestants, headed by a Delegation Leader and a Deputy Leader.

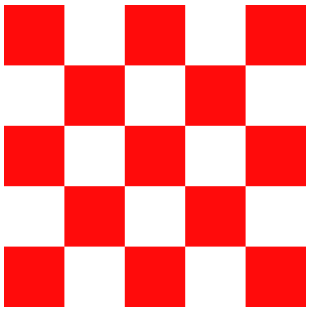
A **Contestant** is a student who was enrolled at a school for secondary education, in the Country they are representing, during the period September to December 2006, and is not older than twenty years on the 1st of July 2007.

The official IOI web site is: http://www.ioinformatics.org
The IOI Secretariat web site is: http://olympiads.win.tue.nl/ioi/

Anexo II - Problemas das duas provas nas IOI

ALIENS

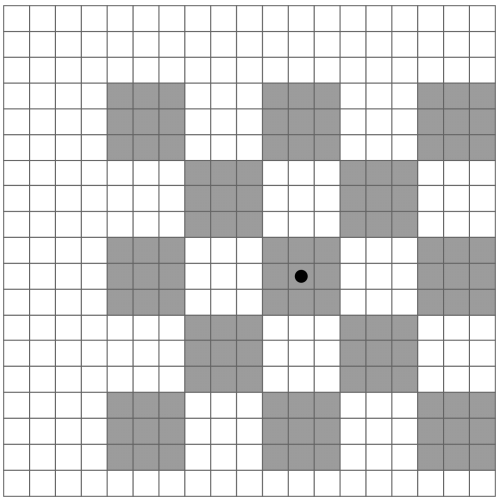
Mirko is a big fan of crop circles, geometrical formations of flattened crops that are supposedly of alien origin. One summer night he decided to make his own formation on his grandmother's meadow. The great patriot that he is, Mirko decided to make a crop formation that would have the shape of the shield part of the Croatian coat of arms, which is a 5×5 chessboard with 13 red squares and 12 white squares.



The chessboard part of the Croatian coat of arms.

Grandma's meadow is a square divided into N×N cells. The cell in the lower left corner of the meadow is represented by the coordinates (1, 1) and the cell in the upper right corner is represented by (N, N).

Mirko decided to flatten only the grass belonging to red squares in the chessboard, leaving the rest of the grass intact. He picked an **odd integer** M ≥ 3 and flattened the grass so that each square of the chessboard comprises M×M cells in the meadow, and the chessboard completely fits inside the meadow.



Example meadow and Mirko's crop formation, with N=19 and M=3.
Cells with flattened grass are shown in gray.
The center of the formation is at (12, 9) and is marked with a black point.

After Mirko went to sleep, his peculiar creation drew the attention of real aliens! They are floating high above the meadow in their spaceship and examining Mirko's crop formation with a simple device. This device can only **determine whether the grass in a particular cell is flattened or not**.

The aliens have found **one cell with flattened grass** and now they want to find the **center cell** of Mirko's masterpiece, so that they may marvel at its beauty. They **do not know the size M** of each square in Mirko's formation.

TASK

Write a program that, given the size N ($15 \leq N \leq 2\,000\,000\,000$) of the meadow, the coordinates (X_0, Y_0) of one cell with flattened grass, and the ability to interact with the alien device, finds the coordinates of the center cell of Mirko's crop formation.

The device may be used at most 300 times in one test run.

INTERACTION

This is an interactive task. Your program sends commands to the alien device using the standard output, and receives feedback from the device by reading from the standard input.

- At the beginning of your program, you should read three integers N, X_0 and Y_0 from the standard input, separated by single spaces. The number N is the size of the meadow, while (X_0, Y_0) are the coordinates of one cell with flattened grass.
- To examine the grass in the cell (X, Y) using the alien device, you should output a line of the form `"examine X Y"` to the standard output. If the coordinates (X, Y) are not inside the meadow (the conditions $1 \leq X \leq N$ and $1 \leq Y \leq N$ are not satisfied), or if you use this facility more than 300 times, your program will receive a score of zero on that test run.
- The alien device will respond with a single line containing the word `"true"` if the grass in cell (X, Y) is flattened and the word `"false"` otherwise.
- When your program has found the center cell, it should output a line of the form `"solution` X_C Y_C`"` to the standard output, where (X_C, Y_C) are the coordinates of the center cell. The execution of your program will be automatically terminated once your program outputs a solution.

In order to interact properly with the grader, your program needs to **flush the standard output** after every write operation; the provided code samples show how to do this.

CODE SAMPLES

Code samples in all three programming languages are available for download on the "Tasks" page of the contest system. The purpose of the samples is only to show how to interact with the alien device; these are not the correct solutions and will not score all points.

GRADING

In test cases worth a total of 40 points, the size M of each of Mirko's squares will be at most 100.

Each test run will have a unique correct answer that will not depend on the questions asked by your program.

EXAMPLE

In the following example, commands are given in the left column, row by row. Feedback from the alien device is given in the second column of the corresponding row.

output (command)	input (feedback)
	19 7 4
examine 11 2	true
examine 2 5	false
examine 9 14	false
examine 18 3	true
solution 12 9	

TESTING

During the contest, there are three ways to test your solutions.

The first way is for you to simulate the alien device manually and interact with your program.

The second way is to write a program which will simulate the alien device. To connect your solution and the device you wrote, you may use a utility called `"connect"`, available for download on the contest system. To use the utility issue a command such as `"./connect ./solution ./device"` from the console (substituting `"solution"` and `"device"` with the names of your two programs). Any additional command-line parameters will be passed on to the device program.

The third way is to use the TEST facility of the grading system to automatically run your solution with a custom test case. When using the facility, the size of the meadow N is limited to 100.

A test case should contain three lines:

- The first line contains the size N of the meadow and the size M of a square in the chessboard;
- The second line contains the coordinates X_0 and Y_0 of one cell of the meadow with flattened grass, which will be given to your program;
- The third line contains the coordinates X_C and Y_C of the center cell of the chessboard.

The grading system will provide you with a detailed log of the execution, including error messages if:

- N doesn't satisfy the constraints;
- M is not an odd integer greater than or equal to 3;
- The crop formation doesn't fit in the meadow;
- The grass in cell (X_0, Y_0) is not flattened.

Here is an example of a valid input file for the test facility. The example corresponds to the figure on the first page.

```
19 3
7 4
12 9
```

Valid input for the test facility.

FLOOD

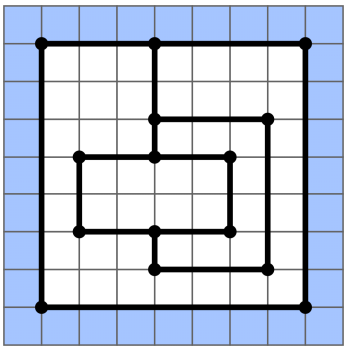
In 1964 a catastrophic flood struck the city of Zagreb. Many buildings were completely destroyed when the water struck their walls. In this task, you are given a simplified model of the city before the flood and you should determine which of the walls are left intact after the flood.

The model consists of N points in the coordinate plane and W walls. **Each wall connects a pair of points and does not go through any other points**. The model has the following additional properties:

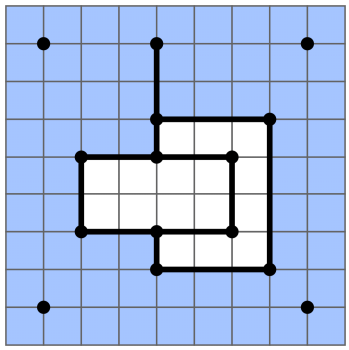
- No two walls intersect or overlap, but they may touch at endpoints;
- Each wall is parallel to either the horizontal or the vertical coordinate axis.

Initially, the entire coordinate plane is dry. At time zero, water instantly floods the exterior (the space not bounded by walls). After exactly one hour, every wall with water on one side and air on the other breaks under the pressure of water. Water then floods the new area not bounded by any standing walls. Now, there may be new walls having water on one side and air on the other. After another hour, these walls also break down and water floods further. This procedure repeats until water has flooded the entire area.

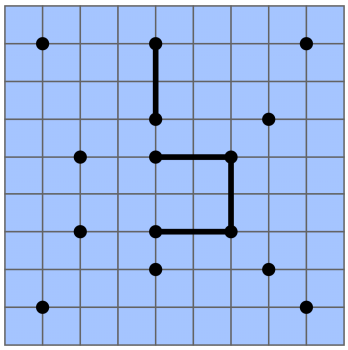
An example of the process is shown in the following figure.



The state at time zero. Shaded cells represent the flooded area, while white cells represent dry area (air).



The state after one hour.



The state after two hours. Water has flooded the entire area and the 4 remaining walls cannot be broken down.

TASK

Write a program that, given the coordinates of the N points, and the descriptions of W walls connecting these points, determines which of the walls are left standing after the flood.

INPUT

The first line of input contains an integer N ($2 \leq N \leq 100\,000$), the number of points in the plane.

Each of the following N lines contains two integers X and Y (both between 0 and 1 000 000, inclusive), the coordinates of one point. The points are numbered 1 to N in the order in which they are given. No two points will be located at the same coordinates.

The following line contains an integer W ($1 \leq W \leq 2N$), the number of walls.

Each of the following W lines contains two different integers A and B ($1 \leq A \leq N$, $1 \leq B \leq N$), meaning that, before the flood, there was a wall connecting points A and B. The walls are numbered 1 to W in the order in which they are given.

OUTPUT

The first line of output should contain a single integer K, the number of walls left standing after the flood.

The following K lines should contain the indices of the walls that are still standing, one wall per line. The indices may be output in any order.

GRADING

In test cases worth a total of 40 points, all coordinates will be at most 500.

In those same cases, and cases worth another 15 points, the number of points will be at most 500.

DETAILED FEEDBACK WHEN SUBMITTING

During the contest, you may select up to 10 submissions for this task to be evaluated (as soon as possible) on part of the official test data. After the evaluation is done, a summary of the results will be available on the contest system.

EXAMPLE

input

```
15
1 1
8 1
4 2
7 2
2 3
4 3
6 3
2 5
4 5
6 5
4 6
7 6
1 8
4 8
8 8
17
1 2
2 15
15 14
14 13
13 1
14 11
11 12
12 4
4 3
3 6
6 5
5 8
8 9
9 11
9 10
10 7
7 6
```

output

```
4
6
15
16
17
```

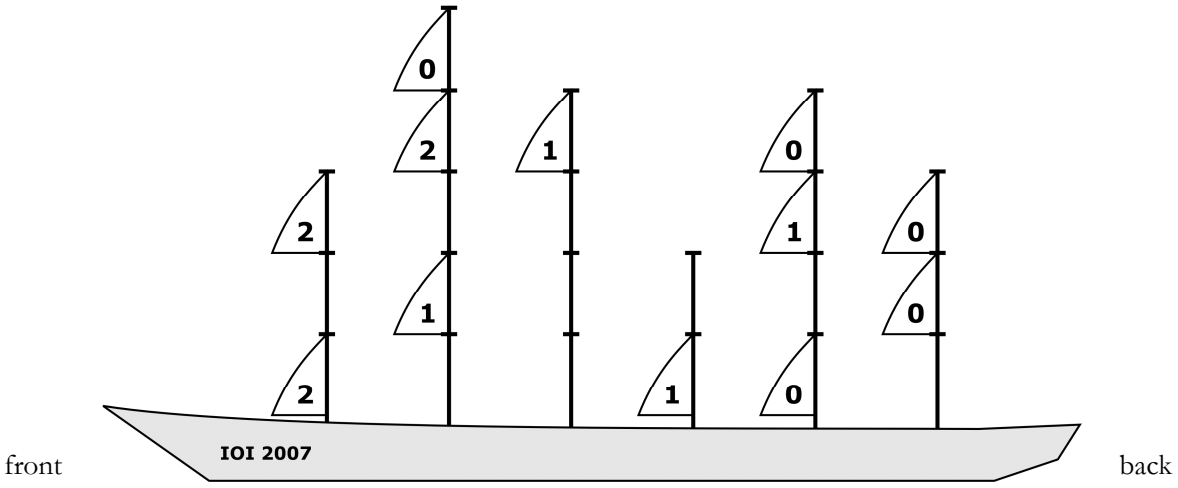
This example corresponds to the figure on the previous page.

SAILS

A new pirate sailing ship is being built. The ship has N masts (poles) divided into unit sized segments – the height of a mast is equal to the number of its segments. Each mast is fitted with a number of sails and each sail exactly fits into one segment. Sails on one mast can be arbitrarily distributed among different segments, but each segment can be fitted with at most one sail.

Different configurations of sails generate different amounts of thrust when exposed to the wind. Sails in front of other sails at the same height get less wind and contribute less thrust. For each sail we define its **inefficiency** as the total number of sails that are **behind** this sail and **at the same height**. Note that "in front of" and "behind" relate to the orientation of the ship: in the figure below, "in front of" means to the left, and "behind" means to the right.

The **total inefficiency** of a configuration is the sum of the inefficiencies of all individual sails.



This ship has 6 masts, of heights 3, 5, 4, 2, 4 and 3 from front (left side of image) to back.
This distribution of sails gives a total inefficiency of 10. The individual inefficiency of each sail is written inside the sail.

TASK

Write a program that, given the height and the number of sails on each of the N masts, determines the **smallest** possible total inefficiency.

INPUT

The first line of input contains an integer N ($2 \leq N \leq 100\,000$), the number of masts on the ship.

Each of the following N lines contains two integers H and K ($1 \leq H \leq 100\,000$, $1 \leq K \leq H$), the height and the number of sails on the corresponding mast. Masts are given in order from the front to the back of the ship.

OUTPUT

Output should consist of a single integer, the smallest possible total inefficiency.

Note: use a 64-bit integer type to calculate and output the result (`long long` in C/C++, `int64` in Pascal).

GRADING

In test cases worth a total of 25 points, the total number of ways to arrange the sails will be at most 1 000 000.

EXAMPLE

input

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

output

```
10
```

This example corresponds to the figure on the previous page.

MINERS

There are **two** coal mines, each employing a group of miners. Mining coal is hard work, so miners need food to keep at it. Every time a shipment of food arrives at their mine, the miners produce some amount of coal. There are three types of food shipments: meat shipments, fish shipments and bread shipments.

Miners like variety in their diet and they will be more productive if their food supply is kept varied. More precisely, every time a new shipment arrives to their mine, they will **consider the new shipment and the previous two shipments** (or fewer if there haven't been that many) and then:

- If all shipments were of the same type, they will produce one unit of coal.
- If there were two different types of food among the shipments, they will produce two units of coal.
- If there were three different types of food, they will produce three units of coal.

We know in advance the types of food shipments and the order in which they will be sent. It is possible to influence the amount of coal that is produced by determining which shipment should go to which mine. Shipments cannot be divided; each shipment must be sent to one mine or the other in its entirety.

The two mines don't necessarily have to receive the same number of shipments (in fact, it is permitted to send all shipments to one mine).

TASK

Your program will be given the types of food shipments, in the order in which they are to be sent. Write a program that finds the **largest total amount of coal** that can be produced (in both mines) by deciding which shipments should be sent to mine 1 and which shipments should be sent to mine 2.

INPUT

The first line of input contains an integer N ($1 \leq N \leq 100\,000$), the number of food shipments.

The second line contains a string consisting of N characters, the types of shipments in the order in which they are to be distributed. Each character will be one of the uppercase letters 'M' (for meat), 'F' (for fish) or 'B' (for bread).

OUTPUT

Output a single integer, the largest total amount of coal that can be produced.

GRADING

In test cases worth a total of 45 points, the number of shipments N will be at most 20.

DETAILED FEEDBACK WHEN SUBMITTING

During the contest, you may select up to 10 submissions for this task to be evaluated (as soon as possible) on part of the official test data. After the evaluation is done, a summary of the results will be available on the contest system.

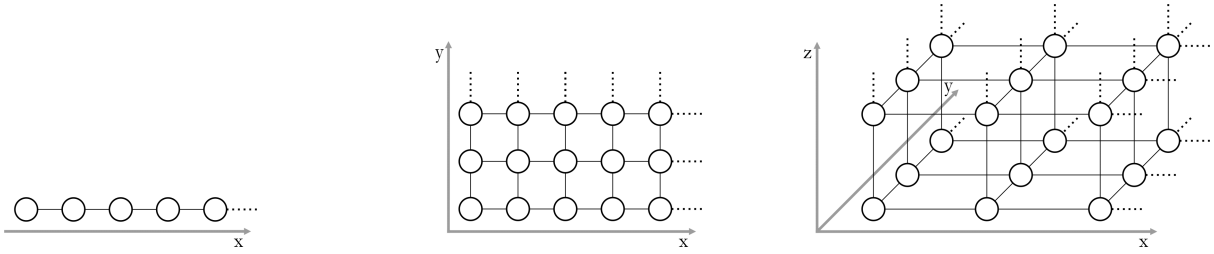
EXAMPLES

input	input
6 MBMFFB	16 MMBMBBBBMMMMMBMB
output	**output**
12	29

In the left sample, by distributing the shipments in this order: mine 1, mine 1, mine 2, mine 2, mine 1, mine 2, the shipments will result in 1, 2, 1, 2, 3 and 3 units of coal produced in that order, for a total of 12 units. There are other ways to achieve this largest amount.

PAIRS

Mirko and Slavko are playing with toy animals. First, they choose one of three boards given in the figure below. Each board consists of cells (shown as circles in the figure) arranged into a one, two or three dimensional grid.



Board 1 Board 2 Board 3

Mirko then places N little **toy animals** into the cells.

The **distance** between two cells is the smallest number of moves that an animal would need in order to reach one cell from the other. In one move, the animal may step into one of the adjacent cells (connected by line segments in the figure).

Two animals can hear each other if the distance between their cells is **at most D**. Slavko's task is to calculate how many **pairs of animals** there are such that one animal can hear the other.

TASK

Write a program that, given the board type, the locations of all animals, and the number D, finds the desired number of pairs.

INPUT

The first line of input contains four integers in this order:

- The board type B ($1 \leq B \leq 3$);
- The number of animals N ($1 \leq N \leq 100\,000$);
- The largest distance D at which two animals can hear each other ($1 \leq D \leq 100\,000\,000$);
- The size of the board M (the largest coordinate allowed to appear in the input):
  - When B=1, M will be at most 75 000 000.
  - When B=2, M will be at most 75 000.
  - When B=3, M will be at most 75.

Each of the following N lines contains B integers separated by single spaces, the coordinates of one toy animal. Each coordinate will be between 1 and M (inclusive).

More than one animal may occupy the same cell.

OUTPUT

Output should consist of a single integer, the number of pairs of animals that can hear each other.

Note: use a 64-bit integer type to calculate and output the result (`long long` in C/C++, `int64` in Pascal).

GRADING

In test cases worth a total of 30 points, the number of animals N will be at most 1 000.

Furthermore, for each of the three board types, a solution that correctly solves all test cases of that type will be awarded at least 30 points.

EXAMPLES

input

```
1 6 5 100
25
50
50
10
20
23
```

output

```
4
```

input

```
2 5 4 10
5 2
7 2
8 4
6 5
4 4
```

output

```
8
```

input

```
3 8 10 20
10 10 10
10 10 20
10 20 10
10 20 20
20 10 10
20 10 20
20 20 10
20 20 20
```

output

```
12
```

Clarification for the leftmost example. Suppose the animals are numbered 1 through 6 in the order in which they are given. The four pairs are:

- 1-5 (distance 5)
- 1-6 (distance 2)
- 2-3 (distance 0)
- 5-6 (distance 3)

Clarification for the middle example. The eight pairs are:

- 1-2 (distance 2)
- 1-4 (distance 4)
- 1-5 (distance 3)
- 2-3 (distance 3)
- 2-4 (distance 4)
- 3-4 (distance 3)
- 3-5 (distance 4)
- 4-5 (distance 3)

TRAINING

Mirko and Slavko are training hard for the annual tandem cycling marathon taking place in Croatia. They need to choose a route to train on.

There are N cities and M roads in their country. Every road connects two cities and can be traversed in both directions. Exactly N−1 of those roads are **paved**, while the rest of the roads are unpaved trails. Fortunately, the network of roads was designed so that each pair of cities is connected by a path consisting of paved roads. In other words, the N cities and the N−1 **paved roads form a tree structure**.

Additionally, each city is an endpoint for **at most 10 roads total**.

A training route starts in some city, follows some roads and ends in the same city it started in. Mirko and Slavko like to see new places, so they made a rule **never to go through the same city** nor **travel the same road twice**. The training route may start in any city and does not need to visit every city.

Riding in the back seat is easier, since the rider is shielded from the wind by the rider in the front. Because of this, Mirko and Slavko change seats in every city. To ensure that they get the same amount of training, they must choose a route with an **even number of roads**.

Mirko and Slavko's competitors decided to **block** some of the unpaved roads, making it **impossible** for them to find a training route satisfying the above requirements. For each unpaved road there is a **cost** (a **positive** integer) associated with blocking the road. It is impossible to block paved roads.

TASK

Write a program that, given the description of the network of cities and roads, finds the **smallest total cost** needed to block the roads so that **no training route exists** satisfying the above requirements.

INPUT

The first line of input contains two integers N and M ($2 \leq N \leq 1\,000$, $N-1 \leq M \leq 5\,000$), the number of cities and the total number of roads.

Each of the following M lines contains three integers A, B and C ($1 \leq A \leq N$, $1 \leq B \leq N$, $0 \leq C \leq 10\,000$), describing one road. The numbers A and B are different and they represent the cities directly connected by the road. If C=0, the road is paved; otherwise, the road is unpaved and C represents the cost of blocking it.

Each city is an endpoint for at most 10 roads. There will never be more than one road directly connecting a single pair of cities.

OUTPUT

Output should consist of a single integer, the smallest total cost as described in the problem statement.

GRADING

In test cases worth a total of 30 points, the paved roads will form a chain (that is, no city will be an endpoint for three or more paved roads).

DETAILED FEEDBACK WHEN SUBMITTING

During the contest, you may select up to 10 submissions for this task to be evaluated (as soon as possible) on part of the official test data. After the evaluation is done, a summary of the results will be available on the contest system.

EXAMPLES

input

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

output

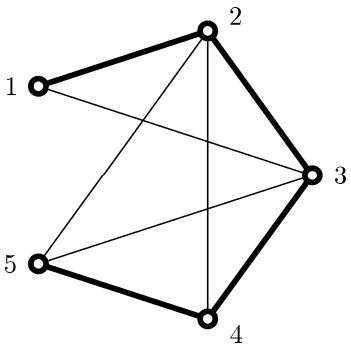
```
5
```

input

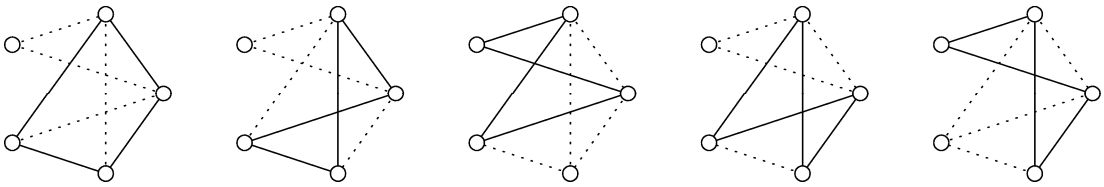
```
9 14
1 2 0
1 3 0
2 3 14
2 6 15
3 4 0
3 5 0
3 6 12
3 7 13
4 6 10
5 6 0
5 7 0
5 8 0
6 9 11
8 9 0
```

output

```
48
```



The layout of the roads and cities in the first example. Paved roads are shown in bold.



There are five possible routes for Mirko and Slavko. If the roads 1-3, 3-5 and 2-5 are blocked, then Mirko and Slavko cannot use any of the five routes. The cost of blocking these three roads is 5.

It is also possible to block just two roads, 2-4 and 2-5, but this would result in a higher cost of 6.

Anexo III - Resultados das IOI

- IOI
  - Introduction
  - History
  - Regulations
- IOI 2007
  - News
  - Organizers
  - Visas
  - Registration
  - Accommodation
  - Fees
  - Schedule
  - Venue
  - IC & ISC
  - March meeting
  - IOI Conference
  - Newsletters
  - Contact
- Competition
  - Rules
  - Tasks
  - Results
  - Online contest
- Location
  - About Croatia
  - About Zagreb
  - Gallery

IOI 2007 RESULTS

Rank	Contestant	Country	A	F	S	M	P	T	Score	Medal
1	Tomasz Kulczyński	Poland	100	100	100	100	74	100	574	Gold
2	Yi Yang	China	100	55	100	100	100	100	555	Gold
3	Hua-yang Guo	China	100	100	100	100	90	35	525	Gold
4	Goran Žužić	Croatia	100	40	100	100	74	100	514	Gold
5	John Pardon	United States	100	100	100	100	80	30	510	Gold
6	Marcin Andrychowicz	Poland	100	100	100	100	74	30	504	Gold
6	Mu Yang	China	100	100	100	100	74	30	504	Gold
8	Qi-feng Chen	China	100	100	23	100	100	60	483	Gold
9	Andrey Kolosov	Russia	100	100	32	100	100	30	462	Gold
9	Richard McCutchen	United States	100	100	32	100	100	30	462	Gold
11	Ilya Razenshteyn	Russia	100	100	25	100	100	30	455	Gold
12	Bo-Jhang Ho	Chinese Taipei	100	100	60	92	50	30	432	Gold
13	Daniyil Neyter	Ukraine	100	100	0	100	100	30	430	Gold
14	Iskren Chernev	Bulgaria	70	100	100	100	54	-	424	Gold
15	Vladislav Epifanov	Russia	100	100	0	100	80	40	420	Gold
15	Zhomart Sadykov	Kazakhstan	100	100	50	100	70	0	420	Gold
17	Bakytzhan Baizhikenov	Kazakhstan	80	55	100	100	80	0	415	Gold
18	Sepideh Mahabadi	Iran	100	40	100	100	64	5	409	Gold
19	Toshiki Kataoka	Japan	100	64	50	100	50	40	404	Gold
20	Henadzi Karatkevich	Belarus	80	55	80	100	56	30	401	Gold
21	Phumchanit Watanaprakornkul	Thailand	100	0	100	100	70	30	400	Gold
22	Daniel Grunwald	Germany	90	73	50	100	50	30	393	Gold
23	Rostislav Rumenov	Bulgaria	80	100	25	92	84	10	391	Gold
24	Christopher Chen	Australia	100	100	32	100	56	0	388	Gold
24	Shang-En Huang	Chinese Taipei	100	100	23	100	50	15	388	Gold
26	Phitchaya Phothilimthana	Thailand	100	73	40	100	74	0	387	Silver
27	Cosmin Gheorghe	Romania	80	55	100	100	50	0	385	Silver
27	Shahar Papini	Israel	80	100	25	100	50	30	385	Silver
29	Jae Hyun Park	Korea	100	73	16	100	90	0	379	Silver

30	**Bogdan-Cristian Tataroiu**	**Romania**	100	55	16	100	100	-	**371**	**Silver**
31	**Michal Danilak**	**Slovakia**	80	64	50	92	60	20	**366**	**Silver**
32	**Timur Abishev**	**Kazakhstan**	100	40	32	100	90	0	**362**	**Silver**
33	**Hung Doan Manh**	**Vietnam**	80	55	25	100	70	30	**360**	**Silver**
34	**Uladzimir Miniailau**	**Belarus**	100	55	53	100	50	0	**358**	**Silver**
35	**Si Young Oh**	**Korea**	30	100	18	100	74	30	**352**	**Silver**
36	**Matteo Boscariol**	**Italy**	100	73	40	100	38	-	**351**	**Silver**
37	**Giovanni Mascellani**	**Italy**	100	100	0	100	50	-	**350**	**Silver**
37	**Jack Murray**	**Australia**	100	100	-	100	50	0	**350**	**Silver**
39	**Saeidreza Seddighin**	**Iran**	100	40	25	100	54	30	**349**	**Silver**
40	**Han-Jay Yang**	**Chinese Taipei**	90	32	40	100	50	30	**342**	**Silver**
41	**Sina Sadeghian Sadeghabad**	**Iran**	100	55	25	100	54	-	**334**	**Silver**
42	**Josef Pihera**	**Czech Republic**	60	73	50	100	50	0	**333**	**Silver**
43	**César Kawakami**	**Brazil**	100	55	25	100	50	0	**330**	**Silver**
43	**Tana Wattanawaroon**	**Thailand**	40	58	53	100	74	5	**330**	**Silver**
45	**Raman Udavichenka**	**Belarus**	40	73	32	100	50	30	**325**	**Silver**
46	**Vladimir Serbinenko**	**Switzerland**	100	32	40	100	52	0	**324**	**Silver**
46	**János Vincze**	**Hungary**	80	40	40	100	54	10	**324**	**Silver**
48	**Andriy Korotkov**	**Ukraine**	80	40	53	100	50	-	**323**	**Silver**
48	**Ye Wang**	**United States**	100	55	0	100	38	30	**323**	**Silver**
50	**Chethiya Abeysinghe**	**Sri Lanka**	100	40	25	100	50	5	**320**	**Silver**
50	**Adrian Airinei**	**Romania**	40	55	25	100	100	-	**320**	**Silver**
52	**Andrei Grigorean**	**Romania**	80	40	25	100	42	30	**317**	**Silver**
53	**Miroslav Klimos**	**Czech Republic**	100	64	16	100	36	-	**316**	**Silver**
53	**Kaan Soral**	**Turkey**	90	32	25	100	54	15	**316**	**Silver**
55	**Igor Čanadi**	**Croatia**	100	35	0	100	50	30	**315**	**Silver**
55	**Vytautas Gruslys**	**Lithuania**	100	35	-	100	50	30	**315**	**Silver**
57	**Vladimir Boza**	**Slovakia**	80	25	25	100	76	5	**311**	**Silver**
58	**Evgeny Fraimovitch**	**Israel**	80	40	40	100	50	0	**310**	**Silver**
58	**Roman Smrz**	**Czech Republic**	80	-	80	100	50	-	**310**	**Silver**
60	**Dan Carmon**	**Israel**	0	100	25	100	50	30	**305**	**Silver**
61	**Hessameddin Akhlaghpour**	**Iran**	40	-	70	100	50	40	**300**	**Silver**
61	**Raziman Thottungal Valapu**	**India**	100	-	53	92	50	5	**300**	**Silver**
63	**Emil Ibrishimov**	**Bulgaria**	100	-	25	100	74	-	**299**	**Silver**
63	**Chen Xing**	**Sweden**	40	64	40	100	50	5	**299**	**Silver**
65	**Chuan Yu Foo**	**Singapore**	60	8	80	100	50	0	**298**	**Silver**
66	**Agustin Santiago Gutierrez**	**Argentina**	100	-	53	92	50	-	**295**	**Silver**
66	**Alexander Kaluzhin**	**Russia**	60	53	8	100	44	30	**295**	**Silver**

66	Taksapaun Kittiakrastien	Thailand	100	40	25	100	30	-	295	Silver
69	Hadar Zvi Itzikowitz	Israel	80	100	8	68	36	0	292	Silver
70	Kestutis Cesnavicius	Lithuania	70	55	32	84	50	0	291	Silver
71	Lik Yau Yip	Hong Kong China	80	32	25	100	50	0	287	Silver
72	Aram Shatakhtsyan	Armenia	60	0	60	100	66	-	286	Silver
72	Yuki Yoshida	Japan	100	0	16	100	50	20	286	Silver
74	Cheuk Ting Li	Hong Kong China	80	-	40	84	74	5	283	Bronze
75	Ralf Kistner	South Africa	70	-	40	100	50	20	280	Bronze
75	Siarhei Tsikhan	Belarus	40	40	40	100	50	10	280	Bronze
77	Pavel Klavik	Czech Republic	70	32	25	100	50	-	277	Bronze
78	Peter Ondruska	Slovakia	60	40	25	100	50	-	275	Bronze
79	Mikloš Homolja	Serbia	60	16	60	84	54	-	274	Bronze
80	Miguel Angel Covarrubias	Mexico	80	16	25	100	50	0	271	Bronze
81	Balázs Szalkai	Hungary	80	-	40	100	50	0	270	Bronze
82	Tomaž Hočevar	Slovenia	100	40	25	52	50	0	267	Bronze
83	Ming Fung Philip Tai	Hong Kong China	100	-	16	100	50	0	266	Bronze
84	Angus McInnes	Australia	100	24	-	100	38	-	262	Bronze
85	András Eisenberger	Hungary	80	-	32	84	42	20	258	Bronze
85	Adrien Jarthon	France	100	-	8	100	50	0	258	Bronze
87	Chih-Cheng Shih	Chinese Taipei	20	43	27	100	66	0	256	Bronze
88	Srivatsan Balakrishnan	India	100	0	25	100	30	-	255	Bronze
88	Miroslav Bogdanović	Serbia	80	0	25	100	50	-	255	Bronze
88	Paolo Comaschi	Italy	70	0	25	100	60	0	255	Bronze
88	Konstantin Lopyrev	Canada	40	27	8	100	80	-	255	Bronze
88	Long Pham	United Kingdom	80	0	25	100	50	0	255	Bronze
93	Marcin Kurczych	Poland	40	15	25	100	74	0	254	Bronze
93	Ruslan Simonenko	Ukraine	80	0	-	100	74	0	254	Bronze
95	Jong Hyuk Lee	Korea	80	-	8	100	64	0	252	Bronze
96	Ioannis Chatzimichos	Greece	20	55	25	100	50	0	250	Bronze
96	Domagoj Kusalić	Croatia	100	-	0	100	50	0	250	Bronze
96	Relja Medić	Croatia	100	-	8	92	50	-	250	Bronze
96	Duc Ngo Minh	Vietnam	20	40	40	100	50	0	250	Bronze
96	Adam Thomsen	Denmark	100	-	-	100	50	-	250	Bronze
101	Wei Quan Lim	Singapore	0	31	53	100	64	0	248	Bronze
102	Massimo Cairo	Italy	100	32	0	100	15	0	247	Bronze
103	Johannes Josi	Switzerland	100	-	16	100	30	-	246	Bronze
103	Viktor Passichenko	Kazakhstan	80	16	-	100	50	0	246	Bronze
105	Jakub Kallas	Poland	30	-	0	100	100	15	245	Bronze
106	Lauri Kenttä	Finland	100	0	25	100	18	-	243	Bronze

106	Andy Tsz Yin Kong	Canada	90	27	8	68	50	-	243	Bronze
106	Hoang Le Bao	Vietnam	80	0	25	100	38	-	243	Bronze
106	Eduardo Ribas	Brazil	100	-	25	100	18	-	243	Bronze
110	Tien Nguyen Hoanh	Vietnam	0	55	16	100	50	20	241	Bronze
111	Lucas Hosseini	France	100	-	0	100	40	0	240	Bronze
112	Dae Wook Kim	Korea	0	64	25	100	50	-	239	Bronze
113	Alexandru Iacob	Moldova	80	-	8	100	50	-	238	Bronze
114	Christos Mantoulidis	Greece	70	-	16	100	50	0	236	Bronze
115	Wa Kai Wong	Macau China	80	0	25	100	30	0	235	Bronze
116	Brian Marshal	Indonesia	40	-	25	100	50	15	230	Bronze
117	Rati Gelashvili	Georgia	20	100	0	68	40	0	228	Bronze
117	Eiichi Matsumoto	Japan	70	16	25	52	50	15	228	Bronze
119	David Tvaltchrelidze	Georgia	80	0	16	100	30	0	226	Bronze
120	Thijs Marinussen	Netherlands	90	40	25	25	38	-	218	Bronze
120	André Rodrigues	Brazil	30	32	-	100	56	-	218	Bronze
122	Martun Karapetyan	Armenia	70	-	8	84	50	5	217	Bronze
123	Mircea Grecu	Moldova	80	-	40	45	50	-	215	Bronze
124	Louis Jachiet	France	20	0	50	92	50	0	212	Bronze
125	Ahmet Ridvan Duran	Turkey	0	36	25	100	50	-	211	Bronze
126	Jia-Han Chiam	Singapore	0	-	40	100	70	-	210	Bronze
126	Ben Strasser	Luxembourg	100	27	8	45	30	0	210	Bronze
128	Peter Calvert	United Kingdom	20	64	40	34	50	0	208	Bronze
128	Riza Oktavian Nugraha Suminto	Indonesia	80	-	53	45	30	-	208	Bronze
130	Magnus Jedvert	Sweden	40	16	-	100	50	-	206	Bronze
131	Ricky Jeremiah	Indonesia	0	-	40	100	50	15	205	Bronze
132	Karol Danutama	Indonesia	40	28	-	100	30	5	203	Bronze
133	Arindam Saha	India	80	0	25	45	50	-	200	Bronze
134	David Benjamin	United States	30	11	8	100	50	-	199	Bronze
134	Otto Ebeling	Finland	100	8	16	45	30	-	199	Bronze
136	Giorgi Nadiradze	Georgia	40	0	8	100	50	0	198	Bronze
137	Rustam Dzhumaniiazov	Ukraine	0	16	31	92	42	15	196	Bronze
138	Dirk Badenhorst Coetzee	South Africa	80	40	0	45	30	-	195	Bronze
139	Stephen Dolan	Ireland	40	17	0	84	50	0	191	Bronze
140	Alexander Mathews	Australia	20	-	40	100	30	-	190	Bronze
141	Ott Tinn	Estonia	50	-	0	100	38	0	188	Bronze
142	Gediminas Liktaras	Lithuania	80	-	32	45	30	-	187	Bronze
143	-	-	0	0	33	100	50	-	183	-
144	-	-	80	-	-	52	50	-	182	-
144	-	-	90	0	-	52	30	10	182	-
144	-	-	0	0	32	100	50	0	182	-
147	-	-	30	0	0	100	50	0	180	-
147	-	-	-	8	25	92	50	5	180	-

147	-	-	60	0	25	45	50	0	**180**	-
150	-	-	10	0	16	100	50	-	**176**	-
150	-	-	-	-	16	100	60	-	**176**	-
152	-	-	100	-	-	45	30	-	**175**	-
153	-	-	80	-	-	61	30	-	**171**	-
154	-	-	20	40	0	76	34	-	**170**	-
154	-	-	70	-	25	45	30	0	**170**	-
156	-	-	-	39	0	100	30	-	**169**	-
157	-	-	10	16	-	92	50	0	**168**	-
157	-	-	20	-	32	100	16	-	**168**	-
159	-	-	90	0	-	45	32	0	**167**	-
159	-	-	30	40	32	45	20	-	**167**	-
159	-	-	0	16	8	100	38	5	**167**	-
162	-	-	20	-	40	76	30	0	**166**	-
162	-	-	0	16	-	100	50	-	**166**	-
164	-	-	70	-	0	43	50	-	**163**	-
164	-	-	0	8	0	100	50	5	**163**	-
164	-	-	30	8	25	45	50	5	**163**	-
167	-	-	30	-	-	100	32	-	**162**	-
167	-	-	80	-	0	52	30	0	**162**	-
169	-	-	40	24	25	59	12	-	**160**	-
169	-	-	20	49	-	45	46	-	**160**	-
171	-	-	0	0	9	100	50	-	**159**	-
172	-	-	0	0	8	100	50	0	**158**	-
173	-	-	-	0	25	100	30	0	**155**	-
174	-	-	10	0	25	100	18	0	**153**	-
175	-	-	80	-	-	45	27	-	**152**	-
175	-	-	30	32	-	60	30	0	**152**	-
177	-	-	-	0	25	76	50	-	**151**	-
178	-	-	-	8	-	92	50	-	**150**	-
179	-	-	0	8	43	45	50	0	**146**	-
180	-	-	70	-	18	27	30	-	**145**	-
181	-	-	60	-	8	25	50	0	**143**	-
182	-	-	60	-	25	25	32	0	**142**	-
183	-	-	0	-	8	100	32	-	**140**	-
184	-	-	40	0	0	76	23	0	**139**	-
185	-	-	40	-	16	52	30	0	**138**	-
186	-	-	30	-	25	52	30	-	**137**	-
187	-	-	40	-	-	45	50	-	**135**	-
187	-	-	40	-	0	45	50	-	**135**	-
187	-	-	0	-	60	45	30	-	**135**	-
187	-	-	40	-	0	45	50	-	**135**	-
191	-	-	20	-	8	76	30	-	**134**	-
192	-	-	100	-	-	-	30	-	**130**	-

193	-	-	-	-	25	52	50	0	**127**	-
194	-	-	50	0	0	45	30	-	**125**	-
194	-	-	30	-	40	9	46	-	**125**	-
194	-	-	10	-	25	45	30	15	**125**	-
197	-	-	40	0	16	45	23	0	**124**	-
198	-	-	40	-	8	45	30	-	**123**	-
199	-	-	0	27	0	45	50	0	**122**	-
200	-	-	10	48	-	45	18	-	**121**	-
201	-	-	20	-	25	45	30	-	**120**	-
201	-	-	100	-	-	0	20	-	**120**	-
201	-	-	20	0	25	45	30	-	**120**	-
201	-	-	30	0	8	52	30	-	**120**	-
201	-	-	-	-	25	45	50	-	**120**	-
206	-	-	0	8	16	45	50	0	**119**	-
207	-	-	10	-	25	52	30	0	**117**	-
208	-	-	20	0	-	45	50	-	**115**	-
208	-	-	40	-	0	45	30	-	**115**	-
208	-	-	0	40	-	45	30	0	**115**	-
211	-	-	-	0	-	76	30	-	**106**	-
211	-	-	40	-	0	36	30	-	**106**	-
213	-	-	40	-	-	45	20	-	**105**	-
214	-	-	10	0	17	45	30	0	**102**	-
215	-	-	-	-	0	68	30	-	**98**	-
216	-	-	40	-	-	7	50	-	**97**	-
217	-	-	10	-	0	36	50	0	**96**	-
217	-	-	20	8	33	0	30	5	**96**	-
219	-	-	20	0	-	45	30	-	**95**	-
219	-	-	40	-	-	25	30	-	**95**	-
219	-	-	0	0	0	45	50	-	**95**	-
222	-	-	40	-	-	16	38	-	**94**	-
223	-	-	0	0	25	45	23	-	**93**	-
224	-	-	0	8	8	45	30	0	**91**	-
225	-	-	30	-	0	45	12	0	**87**	-
226	-	-	40	-	-	45	0	-	**85**	-
226	-	-	20	-	8	34	23	-	**85**	-
228	-	-	-	8	-	45	30	-	**83**	-
228	-	-	0	-	8	45	30	0	**83**	-
230	-	-	0	0	0	52	30	0	**82**	-
230	-	-	-	0	-	52	30	-	**82**	-
232	-	-	30	-	-	36	15	-	**81**	-
233	-	-	0	0	0	27	50	-	**77**	-
234	-	-	-	-	0	45	30	0	**75**	-
234	-	-	-	0	-	45	30	-	**75**	-
234	-	-	-	-	-	45	30	-	**75**	-

234	-	-	0	-	0	45	30	-	**75**	-
234	-	-	0	-	-	45	30	-	**75**	-
234	-	-	0	-	0	45	30	-	**75**	-
240	-	-	0	-	16	27	30	-	**73**	-
241	-	-	0	-	8	45	18	0	**71**	-
242	-	-	0	-	-	52	18	-	**70**	-
243	-	-	0	-	8	45	12	0	**65**	-
243	-	-	-	0	-	45	20	-	**65**	-
245	-	-	0	-	0	34	30	-	**64**	-
245	-	-	0	-	0	34	30	-	**64**	-
247	-	-	30	0	0	-	30	-	**60**	-
248	-	-	0	0	16	43	-	-	**59**	-
249	-	-	0	-	-	27	30	0	**57**	-
250	-	-	0	-	-	26	30	-	**56**	-
250	-	-	20	0	0	18	18	0	**56**	-
252	-	-	0	-	0	25	30	-	**55**	-
252	-	-	-	-	0	25	30	-	**55**	-
252	-	-	0	0	-	25	30	0	**55**	-
252	-	-	0	-	-	25	30	-	**55**	-
256	-	-	20	-	0	0	30	-	**50**	-
257	-	-	0	-	-	18	30	0	**48**	-
257	-	-	0	0	0	18	30	0	**48**	-
259	-	-	10	-	16	9	12	-	**47**	-
260	-	-	0	-	-	45	0	0	**45**	-
261	-	-	40	-	-	0	-	-	**40**	-
262	-	-	0	-	0	9	30	0	**39**	-
262	-	-	0	-	0	9	30	-	**39**	-
264	-	-	0	-	8	0	30	-	**38**	-
265	-	-	0	-	0	36	-	-	**36**	-
266	-	-	20	-	0	9	5	-	**34**	-
267	-	-	-	0	0	0	30	-	**30**	-
267	-	-	0	-	0	-	30	-	**30**	-
267	-	-	-	-	0	0	30	-	**30**	-
267	-	-	0	0	0	0	30	0	**30**	-
271	-	-	-	0	0	25	0	-	**25**	-
271	-	-	0	-	-	25	-	-	**25**	-
271	-	-	0	-	0	25	0	0	**25**	-
274	-	-	0	0	-	9	15	-	**24**	-
275	-	-	-	-	0	0	20	0	**20**	-
276	-	-	0	-	-	18	-	0	**18**	-
276	-	-	-	-	-	18	-	-	**18**	-
278	-	-	0	-	-	-	12	-	**12**	-
279	-	-	0	0	0	9	0	0	**9**	-
279	-	-	-	0	-	9	-	-	**9**	-

279	-	-	0	-	-	9	-	-	**9**	-
282	-	-	0	-	-	-	-	-	**0**	-
282	-	-	-	0	-	-	0	-	**0**	-
282	-	-	0	-	-	0	0	-	**0**	-
282	-	-	0	0	0	0	0	0	**0**	-

Unofficial team

Rank	Contestant	Country	A	F	S	M	P	T	Score	Medal
-	**Filip Pavetić**	**Croatia II**	30	24	32	100	70	-	**256**	**Bronze**
-	**Bruno Rahle**	**Croatia II**	80	0	25	53	30	-	**188**	**Bronze**
-	-	-	0	24	25	84	50	0	**183**	-
-	-	-	40	-	-	52	50	0	**142**	-