



GeOlymp Series 2010

Episode V

#	Problem Name	Time Limit	Memory Limit
A	card	1 sec.	64 MB
B	credit	1 sec.	64 MB
C	sqrt	1 sec.	64 MB
D	tree	1 sec.	64 MB
E	parts	2 sec.	64 MB

## ამოცანა A. "სადებეტო ბარათი"

ნემომ ბანკში აიღო კრედიტი და საჩუქრად მიიღო სადებეტო ბარათი. მისთვის ეს უკვე მე-17 ბარათია. მისი ცრურწმენითა და დაკვირვებებით ბარათი იღბლიანია, თუ მის ნომერში ციფრი 7 სხვა ციფრებზე უფრო ხშირად გვხვდება. ზუსტად რომ ვთქვათ, სხვა ციფრების რაოდენობა ბარათის ნომერში ნაკლები უნდა იყოს ვიდრე 7-იანების რაოდენობა. დაეხმარეთ ნემოს იმის დადგენაში, იღბლიანია თუ არა ახალი ბარათი და შესაბამისად ღირს თუ არა მისი საფულით ტარება.

### შეზღუდვები:

ბარათის ნომერი შედგება ზუსტად 16 ციფრისაგან.

### შემომავალი ფაილის ფორმატი:

**card.in** ფაილის ერთადერთი სტრიქონი შეიცავს ბარათის ნომერს. სტრიქონი მთავრდება ახალ ხაზზე გადასვლის სიმბოლოთი.

### გამომავალი ფაილის ფორმატი:

**card.out** ფაილში გამოიტანეთ TRUE თუ ბარათი ნემოს გაგებით იღბლიანია, ხოლო FALSE წინააღმდეგ შემთხვევაში.

შემომავალი ფაილის მაგალითი (card.in)	გამომავალი ფაილის მაგალითი (card.out)
1234567891234567	FALSE
7777777777123457	TRUE
7777711122233344	FALSE

## ამოცანა B. "კრედიტი"

როგორც წინა ამოცანიდან შევითქვით, ნემომ ბანკიდან აიღო კრედიტი, რომელიც წარმოადგენს  $M$  თანხას თვიური  $K$  პროცენტით. გრაფიკის მიხედვით მან უნდა იხადოს ყოველთვე ფიქსირებული  $N$  თანხა, ხოლო თუ ბოლო თვეს დარჩენილი თანხა ნაკლებია  $N$ -ზე მაშინ გადაიხდის მხოლოდ დარჩენილ თანხას. ბანკთან დადებული ხელშეკრულების მიხედვით, ყოველი თვის ბოლოს ხდება არსებულ დავალიანებაზე  $K$  პროცენტის დარიცხვა და მხოლოდ ამის შემდეგ იხდის ნემო ფიქსირებულ  $N$  თანხას.

ნემო ასევე გამოცდილი ბიზნესმენიც არის და გარკვეული ბიზნეს-დაბანდებიდან  $T$  თვეში მიიღებს მოგებას, რისი საშუალებითაც აპირებს ბანკის ვალის სრულად დაფარვას. მას აინტერესებს  $T$  თვის შემდეგ რამდენი ექნება ბანკის ვალი, რომ წინასწარ განსაზღვროს თავისი ბიუჯეტი და განთავისუფლდეს ვალისაგან.

### შეზღუდვები:

$$0 \leq N \leq M \leq 1000000$$

$$0 \leq K \leq 100$$

$$1 \leq T \leq 12$$

$N$ ,  $M$ ,  $K$  და  $T$  მთელი რიცხვებია.

### შემომავალი ფაილის ფორმატი:

**credit.in** ფაილის ერთადერთი სტრიქონი შეიცავს ოთხ მთელ  $M$ ,  $K$ ,  $N$  და  $T$  რიცხვს.

### გამომავალი ფაილის ფორმატი:

**credit.out** ფაილში გამოიტანეთ ნემოს დავალიანება  $T$  თვის შემდეგ  $1e-2$  (მძიმის შემდეგ ორი ციფრის) სიზუსტით.

შემომავალი ფაილის მაგალითი (credit.in)	გამომავალი ფაილის მაგალითი (credit.out)
1000 10 10 2	1189.00
1000 10 500 12	0.00

## ამოცანა C. "ფესვის ამოღება"

ამოცანის ავტორი დაიღალა ზღაპრების მოფიქრებით და ამიტომ მოკლე და ლაკონური პირობა შემოგთავაზათ.

დათვალეთ შემდეგი გამოსახულების მნიშვნელობა:

$$\text{floor}(\text{sqrt}(1)) + \text{floor}(\text{sqrt}(2)) + \dots + \text{floor}(\text{sqrt}(N)),$$

სადაც  $\text{sqrt}(A)$  არის  $A$ -ს კვადრატული ფესვის მნიშვნელობა, ხოლო  $\text{floor}(B)$  არის უდიდესი მთელი რიცხვი, რომელიც ნაკლებია ან ტოლია  $B$ -ზე.

**შეზღუდვები:**

$$1 \leq N \leq 10000000000000 \ (10^{12})$$

$N$  მთელი რიცხვია.

**შემომავალი ფაილის ფორმატი:**

**sqrt.in** ფაილის ერთადერთი სტრიქონი შეიცავს ერთ მთელ რიცხვ  $N$ -ს.

**გამომავალი ფაილის ფორმატი:**

**sqrt.out** ფაილში გამოიტანეთ მოცემული გამოსახულების შედეგი.

შემომავალი ფაილის მაგალითი (sqrt.in)	გამომავალი ფაილის მაგალითი (sqrt.out)
1	1
12	25

## ამოცანა D. "ხის ბალანსირება"

ნემოს ძალიან უყვარს არაორდინალური ნივთების ყიდვა. ბოლოს აუქციონზე მან იყიდა ქანდაკება, რომელსაც არასრული ორობითი ხის ფორმა აქვს და ჭერში არის დაკიდებული ხის ფუძით.

ერთხელაც როდესაც ნემო საუზმობდა და ტკბებოდა თავის შენაძენით, მის ერთ-ერთ წვეროზე დაკიდა ერთკილოიანი ბურთულა, რის შედეგადაც ხემ განიცადა დეფორმაცია. ბევრი ფიქრის შემდეგ იგი მიხვდა – ხე რომ არ დეფორმირდეს, საჭიროა მისი ყოველი წვეროსათვის მის ქვეხეებში დაკიდებული ბურთულების რაოდენობა იყოს ტოლი. ანუ ყველა წვეროსთვის, რომელსაც ორი შვილი ჰყავს, ამ შვილებში ჩამოკიდებულ ქვეხეებში ბურთულების რაოდენობა ტოლი უნდა იყოს.

დაეხმარეთ ნემოს და დათვალეთ ყოველ დილით დაკიდებული ერთი ბურთულის შემდეგ რა მინიმალური ბურთულების რაოდენობა დასჭირდება, რომ ხე ისევ ბალანსირებული იყოს. ჩათვალეთ, რომ ამ ბალანსირებას ნემო საღამოსვე აკეთებს და მეორე დილას ბურთულის ჩამოკიდებამდე ხე ისევ ბალანსირებულია.

### შეზღუდვები:

ხის წვეროების რაოდენობა  $N$  აკმაყოფილებს  $2 \leq N \leq 10000$ .

ხის სიმაღლე არ აღემატება 60-ს.

ხის ყოველ წვეროს 0, 1 ან 2 შვილი ჰყავს.

წვერეები გადანომრილია 0-დან  $(N-1)$ -ის ჩათვლით.

0 ნომრის მქონე წვერო არის ხის ფუძე.

დღეების რაოდენობა, როდესაც ნემო კიდებს ახალ ბურთულას  $1 \leq M \leq 10000$ .

### შემომავალი ფაილის ფორმატი:

**tree.in** ფაილის პირველ სტრიქონზე მოცემულია  $N$  და  $M$ .

შემდეგი  $(N-1)$  სტრიქონიდან თითოეული შეიცავს ორ მთელ რიცხვს  $A$  და  $B$ , რაც ნიშნავს, რომ  $B$  ნომრის მქონე წვერო  $A$  ნომრის მქონე წვეროს შვილია.

შემდეგი  $M$  სტრიქონიდან თითოეულზე მოცემულია ერთი მთელი რიცხვი, რომელიც მიუთითებს ხის წვეროს, სადაც ნემომ დაკიდა ბურთულა.

**გამომავალი ფაილის ფორმატი:**

**tree.out** ფაილში M დაკიდებული ბურთულიდან თითოეულისთვის გამოიტანეთ ხის დაბალანსებისათვის საჭირო ბურთულების მინიმალური რაოდენობა.

შემომავალი ფაილის მაგალითი (tree.in)	გამომავალი ფაილის მაგალითი (tree.out)
6 3 0 1 0 2 2 3 2 4 3 5 5 1 4	3 1 3

## ამოცანა E. "დეტალები"

როგორც უკვე შევითყვეთ, ნემო გარკვეული ბიზნეს-დაბანდებიდან ელოდება მოგებას, რაც მას დაეხმარება კრედიტის დაფარვაში. მისი მოგება საფრთხის ქვეშ დადგა, ვინაიდან ქარხანას, რომელშიც მას ფული აქვს ჩადებული გაუჭირდა პროექტის დასრულება.

ნემომ გადაწყვიტა მათი დახმარება, მაგრამ ვინაიდან საქმე მართლა რთული აღმოჩნდა და თან მის გადასაჭრელად პროგრამა არის დასაწერი, გადაწყდა რომ იგი ამოცანად მიცემოდეთ ჯეოლიმპის მონაწილეებს.

ამოცანის არსი მდგომარეობს შემდეგში, ქარხანაში არსებობს ორი ტიპის დეტალი. პირველი ტიპის დეტალი არის ტოლფერდა სამკუთხედის ტიპის კბილანების მიმდევრობა. ორი გვერდიგერდ მყოფი სამკუთხედი ერთმანეთისაგან დაშორებული იქნება  $D \geq 0$  მანძილით. იხილეთ ნახ.1 და ნახ.2.

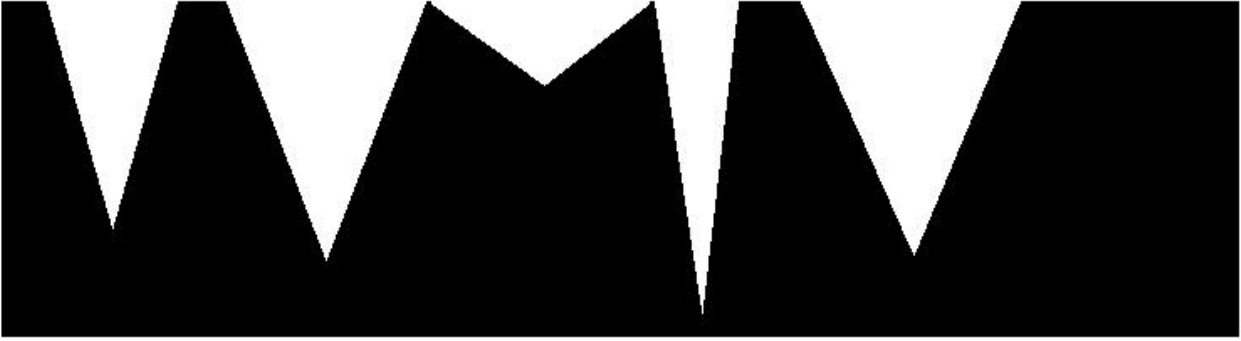


ნახ. 1



ნახ. 2

მეორე ტიპის დეტალი მართკუთხედის ფორმისაა და ზედაპირზე ზოგ ადგილას ამოჭრილი აქვს ტოლფერდა სამკუთხედის ფორმის ფრაგმენტები. იხილეთ ნახ.3.



ნახ. 3

თქვენი ამოცანაა, მოცემული ორი ტიპის დეტალისათვის გაარკვიოთ, თუ რამდენ ადგილას არის შესაძლებელი პირველი დეტალის ჩასმა მეორე დეტალში. პირველი ტიპის დეტალი ჩაისმება მეორეში, თუ მისი კბილანების ფორმები და ზომები ზუსტად ემთხვევა მეორე დეტალის შესაბამის ამოჭრილ ფრაგმენტებს. მაგალითად, დეტალი პირველი ნახაზიდან შესაძლებელია მოთავსდეს დეტალში მესამე ნახაზიდან მხოლოდ ერთ პოზიციაში (დაწყებული მეორე ამოჭრილი სამკუთხედიდან), ხოლო მეორე ნახაზის შესაბამისი დეტალი ვერც ერთხელ ვერ მოთავსდება.

#### **შეზღუდვები:**

პირველი ტიპის დეტალი შედგება  $N \leq 100000$  სამკუთხედისაგან.

მეორე ტიპის დეტალი შედგება  $M \leq 100000$  ამოჭრილი სამკუთხედისაგან.

$$N \leq M$$

თითოეული სამკუთხედის ფუძე და სიმაღლე ორივე ტიპის დეტალში ნაკლებია 1000-ზე. სამკუთხედებს შორის დაშორებებიც 1000-ზე ნაკლებია.

#### **შემომავალი ფაილის ფორმატი:**

**parts.in** ფაილის პირველ სტრიქონი შეიცავს ორ მთელ  $N$  და  $M$  რიცხვს.

შემდეგი  $N$  სტრიქონი შეიცავს სამკუთხედების აღწერას პირველ დეტალში.

პირველი  $N-1$  სტრიქონიდან თითოეული შეცავს სამ მთელ  $A$ ,  $B$ , და  $C$  რიცხვს.

სადაც  $A$  - სამკუთხედის სიმაღლეა,  $B$  - ფუძის სიგრძე, ხოლო  $C$  - მანძილი

შესაბამისი სამკუთხედის ფუძის მარჯვენა წვეროდან შემდეგი სამკუთხედის

ფუძის მარცხენა წვერომდე. ხოლო მე- $N$  სტრიქონი შეიცავს მხოლოდ  $A$ ს და  $B$ ს.

შემდეგი  $M$  სტრიქონიდან თითოეული შეიცავს მეორე დეტალში ჩაჭრილი



სამკუთხედების აღწერას. პირველი M-1 სტრიქონი შეცავს სამ მთელ A, B, და C რიცხვს. სადაც A - სამკუთხედის სიმაღლეა, B - ფუძის სიგრძე, ხოლო C - მანძილი შესაბამისი სამკუთხედის ფუძის მარჯვენა წვეროდან შემდეგი სამკუთხედის ფუძის მარცხენა წვერომდე. ხოლო მე-M სტრიქონი შეიცავს მხოლოდ As და Bs.

**გამომავალი ფაილის ფორმატი:**

**parts.out** ფაილის ერთადერთ სტრიქონში გამოიტანეთ ერთი მთელი რიცხვი, რომელიც წარმოადგენს პოზიციების რაოდენობას, სადაც შეიძლება პირველი დეტალის ჩასმა მეორეში.

შემომავალი ფაილის მაგალითი (parts.in)	გამომავალი ფაილის მაგალითი (parts.out)
3 6 1 2 3 3 4 4 5 1 1 2 3 1 2 3 3 4 4 5 1 2 1 1 1 1 2	1