

**XXII РЕПУБЛИКАНСКА СТУДЕНСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ**  
**Русе, 17 април 2010 г.**

**Задача А. ОСТАТЪЦИ**

Дадени са  $n$  ( $1 < n < 10$ ) различни прости числа  $p_1, p_2, \dots, p_n$ , по-малки от 50. Нека  $r_i$  са неотрицателни цели числа, такива че  $r_i < p_i, i = 1, 2, \dots, n$ . Напишете програма, която намира най-малкото цяло положително число  $x$ , такова че, за всяко  $i = 1, 2, \dots, n$ , остатъкът при делението на  $x$  с  $p_i$  да е равен на  $r_i$ .

**Вход**

Програмата прочита от стандартния вход броя на тестовите примери (не повече от 10). Следват тестовите примери, като за всеки тест данните са записани на един ред в следната последователност:  $n, p_1, p_2, \dots, p_n, r_1, r_2, \dots, r_n$ .

**Изход**

За всеки тест да се изведе на отделен ред в стандартния изход съответната стойност на  $x$ , намерена от програмата.

**Пример:**

**Вход**

```
2
2 2 3 1 2
3 2 3 5 0 1 2
```

**Изход**

```
5
22
```

**XXII РЕПУБЛИКАНСКА СТУДЕНСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ**  
**Русе, 17 април 2010 г.**

**Задача В. НАЙ-КЪСО РАЗСТОЯНИЕ**

Дадени са парабола  $y=ax^2+bx+c$  ( $a \neq 0$ ) и права  $y=kx+n$ . Да се намери дължината на най-късата отсечка, на която единият край лежи на параболата, а другият – на правата.

**Вход**

Програмата трябва да прочете от стандартния вход няколко тестови примера. Всеки тестов пример съдържа на един ред стойностите на  $a, b, c, k, n$  – цели числа от интервала  $[-10, 10]$ .

**Изход**

За всеки тестов пример програмата трябва да изведе на отделен ред на стандартния изход цялата част на търсеното разстояние.

**Пример**

<b>Вход</b>	<b>Изход</b>
1 0 0 1 1	0
1 0 0 2 -1	0
1 0 0 -1 -3	1
2 4 -5 3 -10	1

**XXII РЕПУБЛИКАНСКА СТУДЕНСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ**  
**Русе, 17 април 2010 г.**

**Задача С. РИМСКИ ЧИСЛА**

Дадени са две числа  $a$  и  $b$ , записани с римски цифри и знак за аритметично действие  $op$  (+, -, \*, /). Напишете програма, която пресмята израза  $c = a op b$ .

**Вход**

Програмата трябва да може да обработва няколко примера. От първия ред на стандартния вход се въвежда едно цяло число  $N$  – броят на тестовите примери. Следват по три реда за всеки тестов пример. От първия ред за даден тест се въвежда  $a$ . От втория ред се въвежда  $b$ . От третия ред се въвежда знака за аритметичното действие.

**Изход**

За всеки тестов пример на един ред на стандартния изход програмата трябва да изведе стойността на  $c$ , изписана с главни латински букви.

**Забележка**

Ако при действие изваждане  $a$  и  $b$  са равни, да се изведе празен ред. Стойностите на  $a$  и  $b$  могат да бъдат изписани както с главни, така и с малки латински букви, както и смесено.

**Ограничения**

$$0 < N < 11$$

$$1 \leq a \leq \text{MMMCMXCIX}$$

$$1 \leq b \leq \text{MMMCMXCIX}$$

$$1 \leq c \leq \text{MMMCMXCIX}$$

Пример	Пояснения
<p><b>Вход</b> 2 CxxIIII LXIX + MDXV i *</p> <p><b>Изход</b> CXCI MDXV</p>	<p><b>Римските цифри са <i>непозиционна бройна система</i></b> появила се в древен Рим и използвана и до днес в някои части на света. При нея се използват букви, на които се придават цифрови значения и позволяват изписването на цели числа:</p> <p><b>I</b> или <b>i</b> за едно, <b>V</b> или <b>v</b> за пет, <b>X</b> или <b>x</b> за десет, <b>L</b> или <b>l</b> за петдесет, <b>C</b> или <b>c</b> за сто, <b>D</b> или <b>d</b> за петстотин, <b>M</b> или <b>m</b> за хиляда</p>

**XXII РЕПУБЛИКАНСКА СТУДЕНСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ**  
**Русе, 17 април 2010 г.**

**Задача D. ИЗРАЗ**

Разглеждаме израз, зададен по следния начин  $x_1/x_2/x_3/.../x_n$ , където  $x_i$  са цели положителни числа, за всяко  $i$  ( $1 \leq i \leq n$ ). Изразът се пресмята като деленията се извършват отляво надясно, т.е. стойността на  $1/2/1/2$  е  $1/4$ . За промяна на реда на действията се използват скоби. Например, стойността на  $(1/2)/(1/2)$  е 1.

Напишете програма, която чете набор от изрази от зададения вид и, за всеки от тях, определя дали е възможно чрез използване на скоби да се промени редът на пресмятанията така, че резултатът да е цяло число.

**Вход**

Първият ред на входния файл съдържа цяло число  $d$  – брой на зададените изрази. За всеки израз във входния файл на един ред се задава броят на числата от израза –  $n$ , а на следващите  $n$  реда са записани целите положителни числа  $x_1, x_2, \dots, x_n$ , по едно число на ред.

**Изход**

Изходният файл съдържа  $d$  реда. Ако изразът  $i$  ( $i \leq d$ ) може да бъде преобразуван с помощта на скоби така, че резултатът от пресмятането му да е цяло число, ред  $i$  съдържа числото 1; в противен случай ред  $i$  съдържа числото 0.

**Ограничения**

- $0 < d \leq 50$
- $2 \leq n \leq 10000$
- $1 \leq x_i \leq 1000000000$

**Пример**

Вход	Изход	Пояснения
3	1	Входният файл съдържа три изрза.
4	0	<p>Първият израз включва 4 числа: 1, 2, 1, 2, т.е. изразът е <math>1/2/1/2</math>.</p> <p>При използването на скоби по следния начин: <math>(1/2)/(1/2)</math>, резултатът от пресмятанията ще бъде 1, което е цяло число, затова програмата ще изведе <b>1</b> на първия ред на изходния файл.</p> <p>Вторият израз съдържа 3 числа: 1, 2, 3 или изразът е <math>1/2/3</math>.</p> <p>За този израз не е възможно чрез използване на скоби да се промени редът на действията така, че да се получи като резултат цяло число. Затова програмата ще изведе <b>0</b> на втория ред на изходния файл.</p> <p>Третият израз съдържа 4 числа: 1, 1, 12, 13, т.е. изразът е <math>1/1/12/13</math>.</p> <p>Използването на скоби по следния начин: <math>1/((1/12)/13)</math> ще доведе до получаване на резултат 156, което е цяло число и програмата ще изведе <b>1</b> на третия ред на изходния файл.</p>
1	1	
2		
1		
2		
3		
1		
2		
3		
4		
1		
1		
12		
13		

**XXII РЕПУБЛИКАНСКА СТУДЕНСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ**  
**Русе, 17 април 2010 г.**

**Задача Е. ТЕТРАЕДЪР**

Дадени са координатите на върховете на тетраедър  $A(x_1, y_1, z_1)$ ,  $B(x_2, y_2, z_2)$ ,  $C(x_3, y_3, z_3)$  и  $D(x_4, y_4, z_4)$ . Да се намери обемът на тетраедъра.

**Вход**

Програмата трябва да прочете от стандартния вход няколко тестови примера. Всеки тестов пример съдържа на един ред координатите на върховете на тетраедъра:  $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4$  и  $z_4$ . Координатите са цели числа от интервала  $[-100, 100]$ .

**Изход**

За всеки тестов пример програмата трябва да изведе на отделен ред на стандартния изход търсеният обем във вид на несъкратима обикновена дроб или като цяло число.

**Пример**

Вход	Изход
1 1 0 2 -1 5 3 2 0 4 -3 2	15/2
0 0 0 2 0 0 0 3 0 0 0 5	5

**XXII РЕПУБЛИКАНСКА СТУДЕНСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ**  
**Русе, 17 април 2010 г.**

**Задача F. ЧЕРНА МАТЕРИЯ**

През 2020 г. Институтът за космически изследвания (ИКИ) към Българската академия на науките, след дълги изследвания на тъмната материя открива, че тя съставя космическа мрежа, подобна на паяжина в безкрайното пространство на космоса. Учените в ИКИ правят и друго невероятно откритие – множеството от прави сегменти от тази триизмерна мрежа, които лежат в една равнина и са успоредни съставят паралелни светове, които могат да позволят на човека да пътува изключително бързо през време-пространството. За съжаление, сегментите, които лежат на една права не притежават това свойство.

Освен това, използвайки известния ефект в астрономията, наречен гравитационна леща, учените от ИКИ съставят карта на мрежата от черна материя, в която всеки сегмент от мрежата е представен с целочислените координати на крайните си точки. Сега учените космолози са изправени пред следното предизвикателство – да открият колко паралелни светове има в картата, която са съставили. Разбира се, тази задача не е проблем за един специалист в областта на астроинформатиката.

**Вход**

Входът съдържа много тестови примери. За всеки от тях на първия ред се задава число  $N$  - брой на сегментите от черна материя, лежащи на една равнина. На следващите  $N \leq 10000$  реда са дадени сегментите с  $x$  и  $y$  координатите на техните крайни точки. Координатите са цели числа в интервала  $[-10000, 10000]$ . Разделител между координатите е шпация.

**Изход:**

Броят паралелни светове на нов ред за всеки пример.

**Пример:**

**Вход:**

```
5
0 3 4 3
1 -3 1 -1
14 0 13 5
1 1 3 2
6 0 5 5
```

**Изход**

```
1
```

**XXII РЕПУБЛИКАНСКА СТУДЕНСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ**  
**Русе, 17 април 2010 г.**

**Задача G. СОФТУЕРНА ИНДУСТРИЯ**

Световна софтуерна компания има мрежа от  $N$  софтуерни фирми. От съображения за сигурност срещу промишлен шпионаж те са кодирани с числата  $1, 2, \dots, N$ . Всяка фирма може да извършва всяка една от множество от  $M$  операции. Различните операции също са номерирани с числата  $1, 2, \dots, M$ . Известни са номерата на фирмите, които могат да завършват работата по производство на продукти, както и номерът на фирмата, която единствена може да стартира производството на софтуерен продукт. За готови се считат всички продукти, последната операция на които е извършена в някоя от определените за завършване фирми. За всяка извършена от фирма операция е известен номерът на фирмата, която трябва да приеме за доработка продукта (това може да е и същата фирма). Фирмите, които могат да завършат продукта, могат също да извършват операции в средата на процеса, като предават продукта за доработка на друга фирма.

С цел защита на патентната организация по разработката на продукта е възможно да има поставени и фирми фантоми. Фирмите фантоми са фирми, които извършват операции, но продукта не може да попадне за доработка в тях на никакъв етап от разработката си.

Да се състави програма, която да определи минималния необходим брой фирми, които могат да произведат същите готови продукти и само тях.

**Вход**

От първия ред на стандартния вход се въвежда едно число  $T$  – броя на тестовите примери. Следва описание за всеки тестов пример:

- на първия ред са дадени: броят на софтуерните фирми  $N$ , броят на операциите  $M$  и номерът на фирмата, която започва производството;
- на всеки от следващите  $N$  реда е дадена последователност от  $M$  числа.  $P$ -тото число от  $I$ -тия ред ( $1 \leq I \leq N$ ,  $1 \leq P \leq M$ ) е номерът на фирмата, на която  $I$ -тата фирма предава за доработка продукта, след като е извършила операцията с номер  $P$ ;
- на последния ред са дадени: броят на фирмите, в които софтуерният продукт е в завършен вид и списък с техните номера;

**Изход**

За всеки тестов пример на стандартния изход програмата трябва да изведе търсения минимален брой фирми.

**Ограничения:**

$$1 \leq N \leq 1000$$

$$1 \leq M \leq 100$$

## Пример

Вход	Исход
3	5
7 3 1	5
2 5 7	2
2 3 4	
7 3 4	
6 7 4	
7 6 5	
6 7 5	
6 7 5	
3 2 3 4	
8 2 1	
2 6	
7 3	
1 3	
3 7	
8 6	
3 7	
7 5	
7 3	
1 3	
5 2 1	
4 2	
4 1	
4 3	
4 4	
1 4	
1 4	



**XXII РЕПУБЛИКАНСКА СТУДЕНСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ**  
**Русе, 17 април 2010 г.**

**Задача Н. НОВОСТРОЯЩ СЕ РАЙОН**

В столицата на Патешкото царство строителството нараснало прекалено. Въпреки това, обаче, все още не бил решен жилищния проблем. Ето защо градската управа решила да предостави за застрояване дълга ивица, разположена от едната страна на една от крайните улици в града. За да се избегне застрояването с огромни блокове Градския Патешки Съвет разделил ивицата на парцели, като за всеки парцел бил обявен максималния обем, който е разрешено да се застрои на този парцел. Тъй като настояванията за застрояване на по-големи сгради били много настоятелни, от Управата решили да разрешат обединяване на два, три или най-много четири парцела. При това било разрешено, след като няколко парцела да се обединят, обема, разрешен за застрояване да е равен на най-малкото общо кратно на обемите на обединените парцели.

Сега Жълтото Пате имало за задача да намери такова групиране на парцелите, че да може да се постигне застрояване на възможно най-голям обем. Известно е, че то е наследило пресметливостта на чичо си Скрудж, но в математиката изобщо го няма. Ето защо, то прехвърля своята задача на вас. Напишете програма, която по даден брой на парцелите и разрешените за застрояване обеми на всеки от тях, определя търсеното групиране.

**Вход**

От първия ред на стандартния вход се въвежда цяло число  $K$  – броя на тестовите примери. За всеки тестов пример от първия ред на стандартния вход се въвежда цяло число  $N$  ( $1 \leq N \leq 10000$ ) – броя на парцелите. На втория ред, разделени с интервали, се въвеждат  $N$  цели числа – разрешените за застрояване обеми, които са числа между 1 и 500.

**Изход**

За всеки тестов пример се извеждат, свързани с + номерата на парцелите, участващи в поредната група. Всяка група се извежда на отделен ред. На последния ред се отпечатва максималния обем, който може да се застрои.

**Пример**

<b>Вход</b>	<b>Изход</b>
1	1+2+3+4
9	5+6+7+8
18 6 5 7 35 15 13 26 2	9
	3362

**XXII РЕПУБЛИКАНСКА СТУДЕНСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ**  
**Русе, 17 април 2010 г.**

**Задача I. АНТИКРИЗИСНИ МЕРКИ**

В условията на световна икономическа и финансова криза, правителството на Бурландия предложило на синдикатите дълъг списък с разни и разнообразни мерки за борба с кризата и за пълнене на държавния бюджет. Всяка мярка била оценена с прихода ѝ към бюджета. Синдикатите обаче не били съгласни с дългия списък и поискали списъкът да се съкрати. Помогнете на финансовия министър Симид Дрянов да съкрати списъка така, че в бюджета да влязат максимално количество пари по съкратения списък.

**Вход**

На стандартния вход се задават няколко варианта, всеки вариант на отделен ред. Първото число в реда е  $k$  - дължината на съкратения списък, другите числа на същия ред са приходите по дългия списък.

Дължината на дългия списък е  $n$ , като  $n < 10001$  и  $k < n + 1$ .

Приходите са цели числа в интервала  $[-10000, 10000]$ .

**Изход**

За всеки вариант на отделен ред се отпечатва прихода в бюджета по най-добрия съкратен списък.

Пример:

**Вход**

2 10 15 20 10 25

3 2 9 4

**Изход**

45

15

**XXII РЕПУБЛИКАНСКА СТУДЕНСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ**  
**Русе, 17 април 2010 г.**

**Задача J. МАСЛИНОВО ДЪРВО**

Мики Маус обича маслини. В Олимпийската градина има много маслинови дървета – коренови разбира се. От *корена* на дървото израства неговото *стъбло*, от стъблото – един или повече *клона*, а от всеки клон – нула един или повече други клони. Краят на клон, от който не израстват други клони е *лист*. Стъблото на дървото (ще го наричаме също клон) е означено с 1, а другите клони – с числата от 2 до  $N$ . Всеки клон има дължина, а на всеки лист растат определен брой маслини. Когато се катери или слиза по клон с дължина  $l$  Мики губи  $l$  единици от своята сила, а с всяка изядена маслина възстановява по една единица. Ако силите му се изчерпят докато е на дървото – той умира.

Напишете програма която да определи с какво количество сила Мики трябва да пристигне при дървото, ако иска да изяде всички маслини и да слезе от дървото жив.

**Вход.**

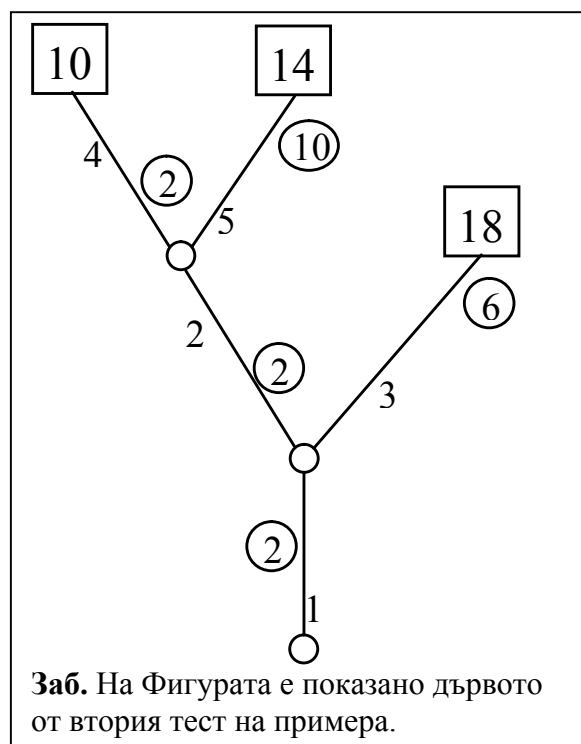
Първият ред на стандартния вход съдържа броя на тестовите. За всеки тест, на първия ред е зададен броят  $N$  на клоните ( $1 \leq N \leq 100$ ).  $i$ -тия от тези редове описва  $i$ -тия клон. Редът започва с дължината  $L$  на клона ( $L$  е цяло,  $1 \leq L \leq 30\,000$ ), последвана от броя  $K$  на клоните израснали от края на  $i$ -тия клон. Ако  $K \neq 0$  – редът завършва с номерата на тези  $K$  клона. Ако  $K = 0$ , тогава клонът завършва с лист. В такъв случай нулата е последвана от броя  $V$  на растящите в листа маслини ( $0 \leq V \leq 30\,000$ ).

**Изход**

За всеки тестов пример, на отделен ред на стандартния изход, програмата трябва да изведе търсения минимален брой единици сила.

**Пример**

Вход	Изход
2	15
8	6
5 3 6 5 7	
5 0 10	
9 0 1	
4 0 19	
11 0 50	
5 2 2 4	
3 2 8 3	
4 0 0	
5	
2 2 2 3	
2 2 4 5	
6 0 18	
2 0 10	
10 0 14	



**XXII РЕПУБЛИКАНСКА СТУДЕНСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ**  
**Русе, 17 април 2010 г.**

**Задача К. КАТЕНИМИ**

*Катеним* е наредена двойка от низове, разделени с точка и такива, че последната буква на първия низ е еднаква с първата буква на втория. Например:

dog.gopher, gopher.rat, rat.tiger, aloha.aloha, arachnid.dog

*Съставен катеним* е редица от три или повече низа, разделени с точки и такива, че всеки два съседни низа в редицата образуват катеним. Например:

aloha.aloha.arachnid.dog.gopher.rat.tiger

Напишете програма, която по зададен речник от низове, съставени от малки латински букви, да намира съставен катеним, който съдържа всеки низ точно веднъж.

**Вход**

На първия ред на стандартния вход е зададен броят на тестовете. Всеки тест започва с ред, съдържащ броя  $N$  на низовете в речника ( $3 \leq N \leq 1000$ ). На всеки от следващите  $N$  реда е зададен по един от низовете; всички низове са различни и с дължина от 1 до 20 малки латински букви.

**Изход**

За всеки тест програмата трябва да изведе на отделен ред на стандартния изход лексикографски най-малкия съставен катеним, съдържащ всяка дума на речника точно един път. Ако съставянето на такъв катеним е невъзможно, тогава програмата трябва да изведе низа \*\*\*.

**Пример**

<b>Вход</b>	<b>Изход</b>
2	aloha.arachnid.dog.gopher.rat.tiger
6	***
aloha	
arachnid	
dog	
gopher	
rat	
tiger	
3	
oak	
maple	
elm	

**XXII РЕПУБЛИКАНСКА СТУДЕНСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ**  
**Русе, 17 април 2010 г.**

**Задача L. NOR**

Както вече знаете, Станчо е много запален по минерална вода и домати. Докато доматите са неговия начин на живот, минералната вода е просто начин за запой. От друга страна Манчо е различен. Той предпочита булеви функции. Станчо не разбира защо някой би предпочитал булевите функции пред минералната вода и доматите, но това е реалността. Манчо е булехолик и има нужда от помощ. Наскоро Станчо измисли начин да помогне на Манчо да откаже булевите функции. За целта е необходимо количество зли програмисти, които биха писали код на някое състезание за без пари...

През годините Манчо е събрал солидно множество от булеви формули. Всяка булева формула е написана на езика C с помощта на неговите функции `and`, `or`, `not`. Освен функциите, той използва и константите 0 и 1, както и променливите `a`, `b` и `c`. Можете да приемете, че тези променливи съдържат стойностите 0 или 1, както и че изброените функции се държат като съответните стандартни булеви операции. Станчо планира да покаже на Манчо, че други неща са по-велики от неговите булеви формули. Той наскоро изобрети станчевите формули. Станчева формула е подобна на булевата, но използва единствено променливите `a`, `b`, и `c`, както и супер яката функция `nor`. Можете да приемете, че `nor(x, y)` е еквивалентно на `not(or(x, y))`. Както и да е, като зъл програмист вие трябва да напишете програма, която получава някои от формулите на Манчо и ги преобразува към възможно най-добрите еквивалентни станчеви формули.

**Вход**

Няколко от булевите формули на Манчо са дадени на стандартния вход. Всяка от тях е написана на отделен ред и съдържа C кодът на формулата без интервали или допълнителни знаци.

**Изход**

За всяка от дадените булеви формули вашата програма трябва да изведе на стандартния изход ред съдържащ C кода на съответната еквивалентна станчева формула. Ако има повече от една еквивалентни станчеви формули изведете тази с най-късия C код. В случай че има няколко най-къси еквивалентни, изведете тази с лексикографски най-малкия код. C кодът не трябва да съдържа интервали или допълнителни знаци.

**Пример**

<b>Вход</b>	<b>Изход</b>
<code>and(a, b)</code> <code>or(b, 0)</code> <code>and(a, not(a))</code> <code>0</code>	<code>nor(nor(a, a), nor(b, b))</code> <code>b</code> <code>nor(a, nor(a, a))</code> <code>nor(a, nor(a, a))</code>