



**Департамент Информатика**  
**Школа „Състезателно програмиране“**  
**СЪСТЕЗАНИЕ, 4 април 2015 г.**

**А. Квадрати**

Напишете програма, която по зададен брой цветове, намира броя на различните квадрати, които може да направите, боядисвайки страните на всеки квадрат с един от тези цветове. Два квадрата са еднакво оцветени, ако единият може да се постави върху другия, така че съвпадащите страни да са боядисани с един и същ цвят.

Стандартния вход съдържа тестове с броя на различните цветове, които могат да бъдат използвани за боядисването на квадратите – естествено число не по-голямо от 32, всяко на отделен ред за поредния тест.

Изведете на стандартния изход по едно число за всеки тест – броят на различно оцветените квадрати, които може да направите с наличните цветове.

<b>Вход</b>	<b>Изход</b>
1	1
2	6
6	231
10	1540



Департамент Информатика  
Школа „Състезателно програмиране“  
СЪСТЕЗАНИЕ, 4 април 2015 г.

**В. Симетрична редица**

Редица от символи е *симетрична*, ако се чете еднакво отляво надясно и обратно. Например, следните редици са симетрични:

123454321  
abssddfddssba

Напишете програма, която по зададена редица намира и извежда минималния брой символи, като и самите символи, които трябва да се добавят към края на редицата, за да стане тя симетрична.

**Вход**

$N$ -те ( $1 \leq N \leq 1050$ ) елементи на редицата от всеки тестов пример се задават на отделен ред на стандартния вход.

**Изход**

За всеки ред от входа програмата трябва да изведе на отделен ред стандартния изход числото  $M$  — минималния брой символи, а на следващия ред самите  $M$  символа, които трябва да се добавят към края на редицата, за да стане тя симетрична. Ако редицата от самото начало е симетрична, то на втория ред не трябва да се извежда нищо.

Вход	Изход
alabala	0
repe	1
alpha21	r 6
	2ahpla



Департамент Информатика  
Школа „Състезателно програмиране“  
СЪСТЕЗАНИЕ, 4 април 2015 г.

**С. Максимум**

По зададено цяло положително число  $n$ , формираме редица по следния начин:

- Ако  $n$  четно, то  $n' = n / 2$
- Ако  $n$  е нечетно, то  $n' = 3n + 1$

При  $n = 5$  получената редицата е 5, 16, 8, 4, 2, 1, 4, 2, 1, ... , а при  $n = 11$  редицата е 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, ...

Така получените редици стигат до 4, 2, 1, след което се получава безкрайно зацикляне. Това все още не е доказано за всички  $n$ , но с помощта на компютър са тествани всички числа до  $5 \times 2^{60}$ . За целите на задачата нека приемем, че когато  $n'$  стане 1, то формирането на редицата е приключило.

Напишете програма, която по зададено  $n$  ( $1 \leq n \leq 100000$ ), намира най-голямото число, измежду всички в получената редица.

На първия ред на стандартния вход ще бъде зададен броят на тестовите примери  $T$  ( $1 \leq T \leq 10000$ ). Всеки от следващите  $T$  реда ще съдържа идентификатора  $K$  на поредния тест, както и самото число  $n$ , разделени с интервал.

За всеки тестов пример извеждайте на отделен ред на стандартния изход  $K$  и търсеното максимално число, разделени с интервал.

Вход	Изход
4	1 1
1 1	2 16
2 3	3 101248
3 9999	4 100000
4 100000	



**Департамент Информатика**  
**Школа „Състезателно програмиране”**  
**СЪСТЕЗАНИЕ, 4 април 2015 г.**

**D. Нумерологията**

Нумерологията е мистично учение, което изучава влиянието на числата върху живота на хората. Например, от първото име на всеки човек зависи каква е мисията на живота му. На всяка буква от името, написано с латински букви съответства число по следния начин:

<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>	<b>I</b>
<b>J</b>	<b>K</b>	<b>L</b>	<b>M</b>	<b>N</b>	<b>O</b>	<b>P</b>	<b>Q</b>	<b>R</b>
<b>S</b>	<b>T</b>	<b>U</b>	<b>V</b>	<b>W</b>	<b>X</b>	<b>Y</b>	<b>Z</b>	
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>

Например, на името ISKRA, числата съответстващи на буквите са последователно I - 9, S - 1, K - 2, R - 9 и A - 1. За да получим ключа, според който се определя мисията ни, трябва да съберем тези числа. Ако сборът е по-голям от 9, събираме цифрите му, докато получим едноцифрено. В примера събираме  $9 + 1 + 2 + 9 + 1 = 22$ ;  $2 + 2 = 4$ . Следователно, името ISKRA има ключ 4. Изключение прави само числото 33. Ако се получи ключ 33, тогава цифрите не се събират и това означава, че мисията ви е специална. За да разберете вашата мисия, трябва да определите какъв е вашият ключ.

Напишете програма, която прочита първото име и пресмята и извежда получения според горното условие ключ.

От стандартния вход се въвежда броя на тестовете, а за всеки от тях по една дума – името, което може да съдържа главни и малки букви от латинската азбука.

За всеки тест на стандартния изход да се изведе едно цяло число – ключът, който се е получил от името. Името може да съдържа най-много 20 символа – главни и малки букви от латинската азбука.

<b>Вход</b>	<b>Изход</b>
10	4
Valeri	9
Stanislav	4
Sonia	9
Karina	2
Mario	3
Vladislav	5
Mitko	5
Krirl	7
Zdravko	6
Samir	



**Департамент Информатика**  
**Школа „Състезателно програмиране”**  
**СЪСТЕЗАНИЕ, 4 април 2015 г.**

**Е. Космическа станция**

Космическа станция има  $N$  секции, номерирани с числата  $1, 2, \dots, N$ , в които са монтирани  $M$  комуникационни устройства. Всяко устройство свързва две секции и е съставено от два модула – по един във всяка от секциите. Модулите на различните устройства в една секция са различни. Движението от една секция в друга може да става само през тези устройства, като две секции може да са свързани с повече от едно устройство. Преминаването от едната от две, свързани с устройство, секции в другата е възможно и в двете посоки и трае  $0.5$  единица време. Придвижването от всяка секция на станцията до всяка друга секция е възможно.

Модулите не работят постоянно. За всеки модул е определено положително цяло  $L$  – дължината на периода на недостъпност на устройството, т.е. модулът е достъпен само в моменти, когато  $K \cdot L$  единици време са изминали от началото на стартиране на системата,  $K=1, 2, 3, \dots$ . Преминаването от една секция в друга е възможно само, когато и двата модула са достъпни. Космонавт стартира системата и иска да премине от секция 1 до секция  $N$  за минималното възможно време. Напишете програма, която решава тази задача.

На първия ред на стандартния вход е зададен броят на тестовите примери. Всеки тестов пример започва с ред с числата  $N$  и  $M$  ( $3 \leq N \leq 100$ ,  $M \leq 1500$ ). Всеки от следващите  $M$  реда описва едно устройство на комуникационната система и съдържа четири цели  $S1$ ,  $L1$ ,  $S2$  и  $L2$ , разделени с интервали.  $S1$  и  $S2$  са номера на свързани с устройство секции, а  $L1$  и  $L2$  са дължините на периодите на недостъпност на съответните модули ( $0 < L1, L2 \leq 1000$ ).

За всеки тестов пример програмата трябва да изведе на отделен ред на стандартния изход минималното необходимо за придвижването от секция 1 до секция  $N$  време, като дробно число с един знак след десетичната точка.

Вход	Изход
1	8.5
4 5	
4 4 3 2	
1 5 3 2	
2 5 3 5	
3 5 4 5	
1 1 2 1	



**Департамент Информатика**  
**Школа „Състезателно програмиране”**  
**СЪСТЕЗАНИЕ, 4 април 2015 г.**

**Г. Цикличен маратон**

Така нареченият цикличен маратон се организира по следните правила. Трасе на маратона е кръгла писта с дължина  $L$ . Една точка на пистата е избрана за Старт. Всеки от  $N$ -те бегачи заема позиция на пистата на някакво разстояние  $D_i$ , измерено по посока на движението на часовниковата стрелка от Старт-Финала и получава стартов номер от 1 до  $N$ , по реда на отдалечаване от Старта. Различните състезатели са на различно разстояние от Старта. След стартовия изстрел всеки състезател започва да бяга по посока на часовниковата стрелка със своя скорост  $S_i$ . Ако състезател застигне бягащия пред него, тогава застигнатият е елиминиран. Състезанието продължава, докато не е възможно никой от останалите да елиминира бягащ пред него. Всички останали на пистата в този момент са победители. Напишете програма run, която да определя реда по който състезателите биват елиминирани.

Първият ред на стандартния вход съдържа броя  $T$  на тестовите примери. Всеки тест започва с ред, на който са зададени положителните цели  $N$  и  $L$  ( $N \leq 50\,000$ ,  $L < 1\,500\,000$ ). На  $i$ -тия от следващите  $N$  реда са зададени характеристиките на  $i$ -тия бегач – разстоянието му до Старта  $D_i$ , цяло число,  $0 \leq D_1 < D_2 < \dots < D_N < L$ , и скоростта му  $S_i$ , дробно число с две цифри след десетичната точка,  $0 < S_i \leq 5$ .

За всеки тестов пример програмата трябва да отпечата на отделен ред на стандартния изход стартовия му номер на поредния елиминиран бегач, по реда на елиминирането. Последният ред на изхода за съответния тест трябва да съдържа низа "Winner(s):", един интервал и списък на победителите, сортирани в нарастващ ред на номерата и разделени с по един интервал.

Вход	Изход
2	2
6 150	3
0 1.75	5
30 0.8	4
60 0.5	6
70 1	Winner(s): 1
120 0.1	Winner(s): 1 2
140 0.9	
2 150	
0 1.75	
140 1.75	



**Департамент Информатика**  
**Школа „Състезателно програмиране“**  
**СЪСТЕЗАНИЕ, 4 април 2015 г.**

### G. Spell Checker

Случвало ли ви се е докато пишете на тъч клавиатурата на смартфона си да объркате думата, която искате да напишете? Всъщност това е много често срещано и за това съществуват коригиращи програми (spell checker), които след като напишете дадена дума ви предлага няколко други, които са „най-близко“, до това което сте написали. Буквите на клавиатурата са разположени по следния начин:

```
qwertyuiop
asdfghjkl
zxcvbnm
```

За целите на задачата, ще считаме, че разстоянието между две думи е равно на сумата от хоризонталното и вертикалното разстояние на всяка от буквите им (Манхатаново разстояние). Например разликата между буквата **w** и **e** е 1, а между **w** и **z** е 3. За простота ще считаме, че написаната дума и предлаганите ви думи ще са с еднаква дължина. Разстоянието между думите **ifpv** и **icpc** е 3.

Първия ред на входа ще съдържа броя на тестовите примери. Първия ред на всеки от тях ще съдържа думата, написана с тъч клавиатурата и броят на предлаганите ви корекции за тази дума  $l$  ( $0 < l \leq 10$ ). Следват  $l$  реда, всеки съдържащ по една от тези думи. Гарантирано е, че дължините на всички думи от поредния тест са еднакви и не надвишават 10000. Също така всички думи ще са съставени само от малки латински букви и между тях няма да има повтарящи се.

За всеки тест трябва да изведете  $l$  реда, всеки от които съдържа по една от предлаганите думи, подредени в нарастващ ред, според разстоянието им до написаната от клавиатурата дума. Ако две думи имат еднакво разстояние, то те трябва да бъдат сортирани лексикографски.

Вход	Изход
2	icpc 3
ifpv 3	gcpc 7
iopc	iopc 7
icpc	edc 0
gcpc	rfv 3
edc 5	wsx 3
wsx	qed 4
edc	plm 17
rfv	
plm	
qed	



**Департамент Информатика**  
**Школа „Състезателно програмиране”**  
**СЪСТЕЗАНИЕ, 4 април 2015 г.**

**Н. Фибоначи по модул**

През 1960 Donald Wall от IBM доказва, че всяко поредно число от редицата на Фибоначи по модул  $m$  формира периодична редица.

Вижте например първите десет елемента от редицата на Фибоначи, както и остатъците им от деленето с 11:

<b>n</b>		<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>F(n)</b>		<b>1</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>5</b>	<b>8</b>	<b>13</b>	<b>21</b>	<b>34</b>	<b>55</b>
<b>F(n) mod 11</b>		<b>1</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>5</b>	<b>8</b>	<b>2</b>	<b>10</b>	<b>1</b>	<b>0</b>

По-нататък редицата от остатъците се повтаря. Ако с  $k(m)$  означим дължината на повтарящата се редицата, то  $k(11) = 10$ .

Ето и някои свойства, които могат да ви се сторят интересни:

- If  $m > 2$ ,  $k(m)$  is even.
- For any even integer  $n > 2$ , there exists  $m$  such that  $k(m) = n$ .
- $k(m) \leq m^2 - 1$
- $k(2^n) = 3 * 2^{(n-1)}$
- $k(5^n) = 4 * 5^n$
- $k(2 * 5^n) = 6n$
- If  $n > 2$ ,  $k(10^n) = 15 * 10^{(n-1)}$

Вашата задача е да изчислявате  $k(m)$  за подадено ви  $m$ .

На първия ред на стандартния вход ще бъде зададен броят на тестовите примери  $T$  ( $1 \leq T \leq 10000$ ). Всеки от следващите  $T$  реда ще съдържа идентификатора  $K$  на поредния тестов пример, както и самото число  $m$  ( $2 \leq m \leq 1000000$ ).

За всеки тест извеждайте на отделен ред на стандартния изход  $K$ , последвано от  $k(m)$ .

Вход	Изход
5	1 6
1 4	2 20
2 5	3 10
3 11	4 15456
4 123456	5 332808
5 987654	





**Департамент Информатика**  
**Школа „Състезателно програмиране”**  
**СЪСТЕЗАНИЕ, 4 април 2015 г.**

**I. Почистване**

Град има  $N$  кръстовища, номерирани с целите числа от 1 до  $N$ . Всяка улица е двупосочна и свързва две различни кръстовища. Някои двойки кръстовища са свързани с една или повече улици. За всеки две кръстовища които не са свързани с улица, има поне един път от улици, по който може да се отиде от едното до другото кръстовище. Кметът на града иска да закупи нови машини за почистването на улиците, като настоява за следната дисциплина на почистване. Всяка машина трябва да има определен за почистване път, който няма общи улици с пътищата на нито една от другите машини, а всяка улица е включена в пътя на точно една от машините. Всяка от машините тръгва да чисти от единия край на своя път и завършва в другия без да минава повече по почистени улици. При това, според законите и дълбокото убеждение на кмета в необходимостта от икономии на общински бюджет, трябва да бъдат поръчани минимален брой машини. Напишете програма, която да определи броя  $M$  на необходимите машини и едно разбиване на града на  $M$  пътя.

На първия ред на стандартния вход ще бъде зададен броят на тестовете. Всеки тест започва с ред, на който е зададен броят на кръстовищата  $N$  ( $3 \leq N \leq 500$ ). На всеки от следващите редове на теста е зададена по една двойка различни цели числа в интервала от 1 до  $N$  – номерата на две кръстовища, свързани с улица. Тестът завършва с ред, на който са зададени две нули. Броят на улиците в теста няма да надхвърля 250 000.

За всеки тестов пример, програмата трябва да изведе на първия ред на стандартния изход намерения минимален брой машини  $M$ . На всеки от следващите  $M$  реда програмата трябва да изведе по един от намерените пътища – списък на върховете в реда, по който се срещат в пътя, завършващ с нула. Ако са възможни няколко различни решения, програмата трябва да изведе кое да е от тях.

Вход	Изход
1	3
6	1 2 3 4 0
1 2	6 5 3 0
2 3	2 5 0
3 4	
3 5	
5 6	
5 2	
0 0	



**Департамент Информатика**  
**Школа „Състезателно програмиране”**  
**СЪСТЕЗАНИЕ, 4 април 2015 г.**

**J. 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**

$$0 < N < 20$$

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

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

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

<p><b>Вход</b></p> 2 CxxIII LXIX + MDXV i *  <p><b>Изход</b></p> CXCI MDXV	<p><b>Пояснения:</b> Римските цифри са <b>непозиционна бройна система</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>
---	---