

# XXII НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг, 29-30 април 2006 г.

Тема за група А (12 клас)

## Задача А1. Повреден часовник

Часовниците често се повреждат. Електронните часовници не правят изключение, като при тях най-честата „повреда“ е, когато част от електронните сегменти на цифрите спрат да светят. Повредените сегменти са лесно видими, но няма никакъв шанс да се разбере дали е включен или изключен някой от повредените сегменти. Така се създава възможност за двузначност: едно показание на часовника би могло да означава два или повече различни часове. За да разрешим проблема, трябва да изчакаме няколко минути и по пътя на логиката да видим кой е верният час.

Нека ни е дадено описание на един повреден електронен часовник: знае се кои сегменти са повредени и какво е текущото показание на часовника. Напишете програма **brkclk**, която пресмята след колко най-малко минути бихме могли да кажем със сигурност колко е часът. Забележете, че можем да твърдим колко (със сигурност) е часът единствено, когато всяка възможност за двузначност е изчезнала.

**Вход.** Часовникът използва стандартните цифри за електронен часовник. На стандартния вход е дадено описание на часовника. То се състои от 4 блокчета от 5x3 символа (5 реда, 3 колонки). Всяко блокче описва една цифра от показанието на часовника. Ето едно примерно блокче, което съответства на показанието от фигурата вдясно:

```

. . .
# . |
. = .
. . #
. = .

```



Тук черните сегменти са включени и работещи, сивият сегмент е изключен, но работещ, белите са неработещи. Възможните цифри, които часовникът може би показва са 5 или 6 (във входа, символите 'I' и '-' означават работещ и изключен сегмент, '#' и '=' - работещ и включен, точките означават неработещ сегмент и запълват останалите позиции).

Цифрите във входа са подравнени хоризонтално и отделени една от друга с два интервала (т.е. всеки ред от входа има по 18 символа и входът съдържа общо 5 реда). Цифрите са подредени в реда ЧЧ:ММ. Винаги се показва някакво валидно време. Часовникът е 24-часов. Показанието му след 23:59 е 0:00 (няма водеща нула на часовете).

**Изход.** Отпечатайте едно единствено число – броя минути, които трябва да изчакате преди да можете да кажете със сигурност колко е часът. Ако не е необходимо да чакате изобщо, отговорът е 0. Ако не можете да кажете колко е часът, без значение колко време ще чакате, отпечатайте числото -1 като отговор.

Пример: Вход	Изход
<pre> . . .      . = .      . . .      . . .   . #      . . #      # . .      # .   . . .      . = .      . . .      . = . . . .      . . .      . . .      # . # . - .      . = .      . . .      . = . </pre>	114

**Пояснение:** Показанието на часовника изглежда, както е показано на фигурата вдясно. Най-лошият случай е, ако времето е било 12:46 или 18:46. Трябва да чакаме 74 минути (до 14:00 или 20:00 съответно) докато показанията станат различни.



**XXII НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА**  
*Национален кръг, 29-30 април 2006 г.*  
**Тема за група А (12 клас)**

**Задача А2. Генератори**

Разглеждаме малките и главните букви от латинската азбука и правила за заместване от следните два вида:

*<главна буква> -> <малка буква>*

*<главна буква 1> -> <главна буква 2><главна буква 3>*

С тези правила могат да се генерират низове по следния начин: започваме с някое правило и получаваме низ, съдържащ една малка или две главни букви (взети от дясната страна на правилото). След това избираме произволна главна буква от низа (ако има такава), избираме правило за което тази главна буква е в лявата му страна (ако има такова правило) и заместваме разглежданата главната буква с дясната страна на избраното правило. Така продължаваме последователно да извършваме замествания, прилагайки всеки път едно произволно взето от дадените правила, докато се получи низ, съдържащ само малки букви. Например, при зададени правила

A->a, B->b, C->AB, A->AA, B->AB, A->CC,

един възможен начин за генериране на низ, в който да останат само малки букви е следният:

B->AB->aB->aAB->aAb->aCCb->aABCb->aaBCb->aabCb->aabABb->aabaBb->aababb.

Напишете програма **gen**, която за даден низ от малки букви и при дадена съвкупност от правила, намира всички правила, от които тръгвайки може да се получи даденият низ.

Програмата трябва да прочете от първия ред на стандартния вход даденият низ (не по-дълъг от 100 знака) от малки латински букви. На следващия ред във входа е даден броят (между 1 и 100) на правилата от първия вид. Следва всяко от тези правила на отделен ред във вида главна буква с долепена малка буква. На следващия ред във входа е даден броят (между 0 и 100) на правилата от втория вид. Следва всяко от тези правила на отделен ред във вида три долепени главни букви.

Програмата трябва да изведе на един ред в стандартния изход левите страни на тези от правилата, започвайки от които, е възможно да се генерира дадения низ. Изведените главни латински букви трябва да са подредени в азбучен ред, без да се повтарят и без разделящи ги интервали. Ако даденият низ не е възможно да бъде генериран, програмата трябва да изведе цифрата 0.

<b>Пример 1. Вход:</b> aababb 2 Aa Bb 4 CAB AAA BAB ACC <b>Изход:</b> BC	<b>Пример 2. Вход:</b> aababb 2 Aa Bb 3 AAA BAB ACC <b>Изход:</b> 0
---	---

**Забележка:** Програмата ще бъде тествана с 10 двойки тестове. Резултатът от всяка двойка тестове е 10 т., само ако и двата теста са успешни, в противен случай се присъждат 0 т. за двойката тестове.

## XXII НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Национален кръг, 29-30 април 2006 г.

Тема за група А (12 клас)

### Задача А3. Вода

Дадени са три мензури А, В и С, всяка с обем  $v$  ml. Мензура А е пълна с течност, а мензури В и С са празни. На мензура А има деления, отбелязващи  $a_1$  и  $a_2$  ml, на мензура В деленията съответстват на  $b_1$  и  $b_2$  ml, а на мензура С – на  $c_1$  и  $c_2$  ml. На всяка стъпка е разрешено да се прелее или цялото количество течност от една мензура в друга, или само част от течността, при условие, че след преливането течността поне в една от двете участващи мензури е точно до някое от деленията. Задачата е с не повече от  $k$  преливания в някоя от мензурите да се получи точно 1 ml течност.

Напишете програма **water**, която намира броя на различните решения.

От първия ред на стандартния вход се въвеждат числата  $v$  ( $1 \leq v \leq 200$ ) – обемът на мензурите и  $k$  ( $1 \leq k \leq 6$ ) – максималният брой ходове. От следващия ред се въвеждат числата  $a_1, a_2, b_1, b_2, c_1, c_2$  – по две деления за всяка от мензурите.

Броят на различните решения да се изведе на стандартния изход.

#### Забележки:

1. В едно решение не се разрешава повторение на конфигурации.
2. След получаването на 1 ml течност в някоя от мензурите, решението завършва и не може да бъде продължавано.

### ПРИМЕР 1

#### Вход

```
20 2
7 10 4 12 3 15
```

#### Изход

	1	2	3	4	5	6
6	20-0-0	20-0-0	20-0-0	20-0-0	20-0-0	20-0-0
	16-4-0	16-4-0	8-12-0	7-13-0	7-0-13	5-0-15
	1-4-15	16-1-3	7-12-1	7-12-1	7-12-1	1-4-15

### ПРИМЕР 2

#### Вход

```
20 3
7 10 4 12 3 15
```

#### Изход

```
72
```