

### Проблем 1. Влада

У далекој земљи Бајтовији, одржали су се редовни парламентарни избори. Након што је изборна комисија прогласила резултате, потребно је формирати владу. Свакој од странака које су прешле цензус је додељен одређени број места за посланике у парламенту. Владу може формирати једна странка или коалиција странака, с тим што укупан број посланика које странка, односно коалиција, поседује мора бити виши од половине свих места у парламенту. Компанија која се бави предвиђањима политичких догађаја жели да зна на колико се начина може формирати постизборна коалиција.

**Улаз.** (Улазни подаци се налазе у датотеци `vlada.in`) У првом реду датотеке се налази природан број  $N$ , број странака које су прешле цензус. У наредних  $N$  редова се налази  $N$  бројева (у сваком реду по један), сваки показује колико је која странка добила места у скупштини.

**Издаз.** (Издазне податке уписати у датотеку `vlada.out`) У први и једини ред датотеке треба исписати број начина да се формира владајућа коалиција.

#### Ограничења.

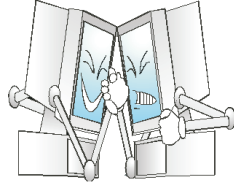
- $1 \leq N \leq 20$ ,
- бројеви места које су странке добиле у парламенту нису већи од 200,
- временско ограничење за извршавање програма је 1s.

#### Пример 1.

<code>vlada.in</code>	<code>vlada.out</code>
4	7
10	
20	
30	
40	

#### Пример 2.

<code>vlada.in</code>	<code>vlada.out</code>
2	2
50	
51	



## Проблем 2. Бирачки списак

Изборна комисије општине Бајтоград је добила допис од Републичке Изборне Комисије о новом начину распоређивања бирача, по коме бирачи добијају свој број који одређује бирачко место на основу њихове позиције у сортираном бирачком списку за целу општину. Нажалост, овај допис им је стигао недељу дана пре избора након што су они већ послали позиве на гласање са бројевима додељеним по старом списку који није био сортиран већ су бирачи дописивани и брисани оним редом којим су добијали и губили бирачко право. Да би уштедели, чланови ОИК-а су се сетили да не морају свима да пошаљу нове позиве већ само оним бирачима чији се број у старом и новом списку разликује. Помозите им да израчунају колико позива треба да пошаљу.

**Улаз.** (Улазни подаци се налазе у датотеци `spisak.in`) Улаз представља стари списак бирача. У првом реду се налази број бирача  $n$ , а у следећих  $n$  редова се налазе имена и презимена бирача, по једно у сваком реду. Бирачу у  $i$ -том реду додељен је број  $i$ . Имена и презимена су састављена од малих слова енглеског алфабета и раздвојени тачно једним размаком.

**Илаз.** (Илазне податке уписати у датотеку `spisak.out`) У првом реду излазне датотеке исписати број бирача којима треба послати нове позиве.

### Ограничења.

- број бирача је мањи од 1000,
- дужина имена и презимена је мања од 25 слова,
- временско ограничење за извршавање програма је 1 s.

**Напомена.** Нови списак је сортиран лексикографски, прво по именима па по презименима. Ако су  $A$  и  $B$  неке две особе са списка онда се одређује прво место на коме се разликују њихова имена (односно прво место на коме се разликују презимена, ако су имена иста). Ако су  $s_A$  и  $s_B$  слова која се налазе на тим местима у речима  $A$  и  $B$ , редом, онда се провери које је од та два слова пре у енглеском алфabetу. За ону реч у којој се налази слово које је пре у алфabetу, кажемо да је *мања* и стоји пре друге (*веће*) речи у сортираном списку. Дозвољено је постојање више особа са истим именом и презименом.

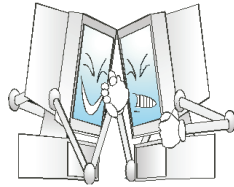
### Пример 1.

<code>spisak.in</code>	<code>spisak.out</code>
4	2
ana popovic	
nikola peric	
nemanja tomic	
marko djordjevic	

### Пример 2.

<code>spisak.in</code>	<code>spisak.out</code>
9	5
ana popovic	
nemanja tomic	
marko djordjevic	
nikola peric	
marko djordjevic	
ana popovic	
nikola peric	
sanja popovic	
marko djordjevic	

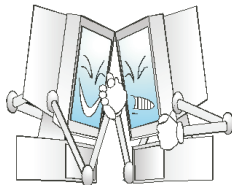
Окружно такмичење  
из информатике  
31. март 2007.



Проблем 2.  
Бирачки списак

---

**Објашњење.** Бирачи којима се не шаљу нови позиви су: ana popovic (број 1), marko djordjevic (број 3), marko djordjevic (број 5) и nikola peric (број 7). Приметимо да би овај списак након лексикографског уређивања имао следећи изглед: ana popovic, ana popovic, marko djordjevic, marko dordjevic, marko djordjevic, nemanja tomic, nikola peric, nikola peric, sanja popovic.



### Проблем 3. Гаус

Професор Ђурић у последње време има великих проблема са својим несташним ђацима. Да би их смирио, одлучио је да им да, на први поглед, много тежак задатак. У прошлости су се често примењивале такве мере смиривања, и омиљен задатак је био сабирање првих 1000 бројева. Али, откако је мали Гаус нашао начин да брзо израчуна тај збир, професори су морали да промене задатак. И тако је проф. Ђурић смислио следеће: даће ђацима не-негативне целе бројеве  $A$  и  $B$ , и тражиће од њих да му кажу колико бројева из интервала  $[A, B]$  (тј. сви бројеви већи или једнаки од  $A$  и мањи или једнаки од  $B$ ) има паран збир цифара (нпр, збир цифара броја 1234 је  $1 + 2 + 3 + 4 = 10$ , дакле паран број). Међутим, међу ђацима се налази и мали Драганче који, попут малог Гауса, жели да што пре реши тај задатак и настави да прави проблеме проф. Ђурићу. Како Драганче није успео да нађе решење задатка, помоћ је потражио од његових другова, младих програмера.

**Улаз.** (Улазни подаци се налазе у датотеци `gaus.in`) У првом реду текстуалне датотеке записани су бројеви  $A$  и  $B$ , одвојени једним размаком.

**Изназ.** (Изназне податке уписати у датотеку `gaus.out`) У излазну датотеку исписати само један број - колико има бројева из интервала  $[A, B]$  таквих да им је збир цифара паран број.

#### Ограничења.

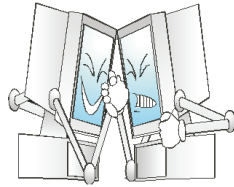
- $0 \leq A \leq B \leq 2^{30}$ ,
- временско ограничење за извршавање програма је 1 s.

#### Пример 1.

<code>gaus.in</code>	<code>gaus.out</code>
5 15	5

#### Пример 2.

<code>gaus.in</code>	<code>gaus.out</code>
16 20	3



#### Проблем 4. Зецови

На пољани се налази  $n$  зечева и  $n$  зечица. Сви они стоје дуж једне линије тако да сви зечеви гледају десно низ линију, а све зечице лево низ линију. Они би желели да се поделе у парове тако да у сваком пару буде једна зечица и један зец који могу међусобно да се виде. Зец види зечицу ако је она било где десно од њега, а зечица види зеца ако је он било где лево од ње. Од вас се тражи да избројте на колико начина они то могу урадити тако да се сваки зец односно зечица нађе у тачно једном пару. Пошто тај број може бити веома велик, нађите само колики остатак даје тај број при дељењу са 10007.



**Улаз.** (Улазни подаци се налазе у датотеци `zecovi.in`) У првом реду записан је број  $n$ . У другом реду записано је тачно  $2n$  симбола (без размака) од којих има  $n$  знакова `>` који означавају зечеве (гледају десно) и  $n$  знакова `<` који означавају зечице (гледају лево).

**Издаз.** (Издазне податке уписати у датотеку `zecovi.out`) У првом реду исписати само један број – остатак при дељењу са 10007 броја могућих начина да се зечеви поделе у парове.

#### Ограничења.

- $1 \leq n \leq 100000$ ,
- временско ограничење за извршавање програма је 1 s.

#### Пример 1.

```
zecovi.in                               zecovi.out
>>><<<                                  4
```

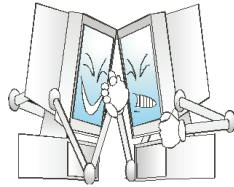
**Објашњење.** Ако зечеве и зечице означимо бројем који одговара позицији на којој се налазе, четири могућа начина формирања парова су:

```
{(1, 6), (2, 3), (4, 5)},
{(1, 3), (2, 5), (4, 6)},
{(1, 3), (2, 6), (4, 5)} и
{(1, 5), (2, 3), (4, 6)}
```

#### Пример 2.

```
zecovi.in                               zecovi.out
8                                          292
>>>>>>>><<<<<<<<<<<<
```

**Објашњење.** Укупан број начина је 40320, што при дељењу са 10007 даје остатак 292.



### Проблем 5. Филмови

Како се мали Драганче из трећег задатка није прославио у математици и програмирању, родитељи су му смањили џепарац. Зато је он решио да искористи своју велику колекцију DVD филмова и одлучио да продаје филмове по ниским ценама. Сваки филм се налази на једном DVD-ју, а на Драганчетовом хард диску може стати највише  $k$  филмова. Процедура резања је следећа: уколико се тражени филм налази на хард диску, Драганче одмах почиње са резањем; у супротном он мора да нађе одговарајући DVD и пресними га на хард диск. Ако на диску нема слободног простора, он мора да обрише неки филм. Тражење DVD-ја и преснимавање изискује пуно времена, и зато Драганче жели да смањи тај број. На почетку је његов диск празан.

Он је направио списак поручених филмова и зна тачно редослед  $n$  купаца који долазе да насниме омиљени филм. Драганче је успео да минимизира број пребацивања филмова на HDD (а самим тим и чекање купаца). Да ли и ви можете да израчунате колико ће најмање пута Драганче ипак морати да пресними неки филм на хард диск?

**Улаз.** (Улазни подаци се налазе у датотеци `filmovi.in`) У првом реду датотеке се налазе два природна броја  $n$  и  $k$ . Број  $n$  представља број наручених филмова, а број  $k$  је број филмова који може да стане на хард диску. У следећих  $n$  редова налазе се редни бројеви филмова  $a[i]$  које купци узимају, поређани по времену доласка.

**Издаз.** (Издазне податке уписати у датотеку `filmovi.out`) У првом и једином реду штампати минималан број преснимавања DVD-ја на хард диск.

#### Ограничења.

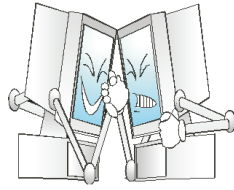
- $1 \leq n \leq 10000$ ,
- $1 \leq k \leq 500$ ,
- $1 \leq a[i] \leq 10000$ ,
- меморијско ограничење 64 kB,
- временско ограничење за извршавање програма је 1 s.

**Начин бодовања:** Број поена се рачуна независно за сваки тест пример. Укупан број тестова је 20 и сваки доноси од 0 до 5 бодова. Претпоставимо да је оптимално решење неког примера број  $OPT$ , а решење такмичара  $SOL$ . Ако је  $SOL < OPT$  или  $SOL > 1.20 \times OPT$ , такмичар не добија поене за дати тест пример. Ако је  $OPT = SOL$ , такмичар добија свих 5 поена. Ако је  $OPT < SOL \leq 1.05 \times OPT$ , добија се 4 поена. Ако је  $1.05 \times OPT < SOL \leq 1.10 \times OPT$ , добија се 3 поена. Ако је  $1.10 \times OPT < SOL \leq 1.15 \times OPT$ , добија се 2 поена. Ако је  $1.15 \times OPT < SOL \leq 1.20 \times OPT$ , добија се 1 поен.

#### Пример 1.

<code>filmovi.in</code>	<code>filmovi.out</code>
5 2	3
1	
2	
2	
4	
1	

**Објашњење.** Како је хард диск на почетку празан, Драганче мора да пресними филм са редним бројем 1. Затим, мора да пресними филм број 2. Следећи купац наручује филм који се већ налази на хард диску, тако да га Драганче одмах нареже. Наредни купац тражи филм 4, тако да Драганче брише филм број 2 и преснимава преко филм број 4. Последњи филм који се тражи је број 1, тако да Драганче не мора да га тражи, јер се на хард диску налазе филмови 4 и 1.



**Пример 2.**

```
filmovi.in
10 3
2
3
2
1
5
2
4
5
3
2
```

```
filmovi.out
6
```