

## 1. Koolimine

1 sekund 20 punkti

Ühes linnas on  $K$  kooli ja iga kooli kohta on teada, mitu õpilast see kool sügisel vastu võtta saab. Linnas on ka  $N$  last, kes peaks sügisel kooli minema, ja iga lapse kohta on teada, millisesse kooli ta minna tahaks.

Kirjutada programm, mis leiab nende andmete põhjal, kui palju on selliseid lapsi, kes ei saa minna soovitud kooli.

**Sisend.** Tekstifaili `kmsis.txt` esimesel real on koolide arv  $K$  ( $1 \leq K \leq 100$ ) ja laste arv  $N$  ( $1 \leq N \leq 10\,000$ ). Koolid on nummerdatud  $1 \dots K$ .

Faili teisel real on  $K$  tühikutega eraldatud täisarvu: koolide mahutavused  $M_i$  ( $1 \leq M_i \leq 1\,000$ ) koolide numbrite järjekorras.

Faili kolmandal real on  $N$  tühikutega eraldatud täisarvu: iga lapse eelistatud kooli number.

**Väljund.** Tekstifaili `kmval.txt` ainsale reale väljastada nende laste arv, kes ei mahu soovitud kooli.

Näide.	<code>kmsis.txt</code>	<code>kmval.txt</code>
	2 5	1
	3 3	
	1 2 2 2 2	

Linnas on kaks kooli ja viis last. Üks laps soovib minna 1. kooli ja ülejäänud 2. kooli. Kuna 2. kooli mahub ainult kolm last, peab üks sinna kooli sooviv laps minema 1. kooli.

## 2. Arvujada redutseerimine

1 sekund

30 punkti

Nimetame täisarvude jada redutseerimiseks seda, kui jagame jada lõikudeks, mis koosnevad ühe väärtuse kordamisest, ja asendame iga lõigu selle pikkusega, nagu näha järgnevates näidetes:

1 1 2 1 1	1 1 2 2 3 3 3 3 2 2 2 7
2 1 2	2 2 4 3 1
1 1 1	2 1 1 1
3	1 3
1	1 1
	2
	1

Kirjutada programm, mis leiab, mitme redutseerimise järel jõuame antud jadast jadani, mille ainus element on 1.

**Sisend.** Tekstifaili `arsis.txt` esimesel real on jada elementide arv  $N$  ( $1 \leq N \leq 1000$ ) ja teisel real  $N$  tühikutega eraldatud täisarvu: jada elemendid  $A_i$  ( $0 \leq A_i \leq 1000$ ).

**Väljund.** Tekstifaili `arval.txt` ainsale reale väljastada 1-jadani jõudmiseks vajalike redutseerimiste arv.

<b>Näide.</b>	<code>arsis.txt</code>	<code>arval.txt</code>
	5	4
	1 1 2 1 1	

<b>Näide.</b>	<code>arsis.txt</code>	<code>arval.txt</code>
	12	6
	1 1 2 2 3 3 3 3 2 2 2 7	

### 3. Kaardilugemine

1 sekund 50 punkti

$N \times M$  pikslit koosneval kaardil on naaberrigid tähistatud erinevate värvidega, kuid riigid, mis ei ole kõrvuti, võivad olla ka sama värvi.

Täpsemalt kuuluvad kaks kokkupuutuvat pikslit kindlasti samasse riiki, kui nad on sama värvi ja neil on ühine serv. Kui samavärvilistel pikslitel on ainult ühine nurk, siis võivad nad olla ka erinevates riikides.

Lisaks on teada, et kaardil pole mitmest eraldiseisvast osast koosnevaid riike.

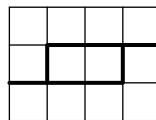
Kirjutada programm, mis leiab kaardil olevate riikide arvu ja riigipiiride kogupikkuse.

**Sisend.** Tekstifaili `klsis.txt` esimesel real on kaardi ridade arv  $N$  ( $1 \leq N \leq 300$ ) ja veergude arv  $M$  ( $1 \leq M \leq 300$ ). Järgmisel  $N$  real on igäihel  $M$  tühikutega eraldatud täisarvu  $K_{ij}$  ( $0 \leq K_{ij} \leq 100$ ): pikslite värvikoodid ridade kaupa.

**Väljund.** Tekstifaili `klval.txt` esimesele reale väljastada kaardil olevate riikide arv ja teisele reale riigipiiride kogupikkus. Riigipiiridena arvestada ainult kaardil nähtavaid riikidevahelisi jooni, kaardi servi piirideks mitte lugeda.

<b>Näide.</b>	<code>klsis.txt</code>	<code>klval.txt</code>
	3 4	3
	1 1 1 1	8
	1 2 2 3	
	3 3 3 3	

Kaardil näha olevad riigipiirid on alloleval joonisel tähistatud rasvasema joonega.



<b>Näide.</b>	<code>klsis.txt</code>	<code>klval.txt</code>
	3 4	3
	1 2 2 1	6
	1 2 2 1	
	1 2 2 1	

**Hindamine.** Selles ülesandes hinnatakse eraldi riikide arvu ja piiride pikkuse leidmist. Programm, mis ei leia riikide arvu, peab faili esimesele reale väljastama arvu  $-1$ . Programm, mis ei leia piiride pikkust, peab faili teisele reale väljastama arvu  $-1$ . Ainult ühe arvu väljastamine on viga ja sellise väljundi eest punkte ei anta.

## 1. Elektrimasinad

1 sekund 30 punkti

Ühes kodus on vaja ajavahemikul kell 00:00–24:00 ära teha  $N$  majapidamistööd. Iga töö jaoks on kindla võimsusega kodumasin ja töö tegemiseks kulub kindel aeg. Mitut tööd võib teha ka samaaegselt, kuid iga alustatud töö tuleb ära teha ilma seda vahepeal katkestamata.

Kirjutada programm, mis saab elektrienergia hinna ööpäeva igas tunnis ja leiab minimaalse kulu, millega on võimalik kõik tööd tehtud saada.

**Sisend.** Tekstifaili `emsis.txt` esimesel real on tööde arv  $N$  ( $1 \leq N \leq 100$ ). Järgmisel  $N$  real on tööde andmed. Igal real on töö tegemiseks kuluv aeg tundides (täisarv  $0 \dots 3$ ) ja minutites (täisarv  $0 \dots 59$ ) ning masina võimsus kilovattides (positiivne reaalarv, mis ei ületa 10). Faili viimasel real on 24 reaalarvu: ühe kilovatt-tunni elektrienergia hind kell 00:00–01:00, 01:00–02:00, ..., 23:00–24:00 (positiivsed reaalarvud, mis ei ületa 10).

**Väljund.** Tekstifaili `emval.txt` ainsale reale väljastada üks reaalarv: minimaalne rahasumma, mis tuleb elektrienergiale kulutada, et kõik tööd südaööks tehtud saada. Vastus ei tohi täpsest väärtusest erineda rohkem kui 0,01 võrra.

**Näide.**

<code>emsis.txt</code>	<code>emval.txt</code>
2	15.00
1 0 3.0	
2 0 1.5	
5 5 5 5 2 5 3 3 5 5 5 5 5 5 5 5 5 5 5 5 5 5	

Esimese töö tegemiseks kulub 3 kW võimsusega masinal täpselt üks tund. Seda on kõige odavam teha kell 04:00–05:00, mil kWh hind on 2.00 ja töö tegemiseks kulub 6.00 väärtuses energiat.

Teise töö tegemiseks kulub 1,5 kW võimsusega masinal täpselt kaks tundi. Seda on kõige odavam teha kell 06:00–08:00, mil kWh hind on 3.00 ja töö tegemiseks kulub 9.00 väärtuses energiat.

## 2. Bitcoin

1 sekund 30 punkti

Bitcoin on rahasüsteem, kus virtuaalseid münte hoitakse virtuaalsetes rahakottides. Igal rahakotil on 168-bitine täisarvuline aadress.

Tekstina esitamiseks kodeeritakse neid aadresse 58-ndsüsteemis järgmise algoritmi alusel (kus `aadress` on muutuja, milles alguses on aadress täisarvuna; `tulemus` on muutuja, millesse lõpuks tekib aadressi esitus tekstina; `div` on täisarvuline jagamine; `mod` on jäägi leidmine):

```
tulemus = ""
kood[0..57] = "123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
ajut = aadress
while ajut > 0
    jaak = ajut mod 58
    ajut = ajut div 58
    tulemus = kood[jaak] + tulemus
ajut = aadress
while ajut 8 vasakpoolsemat bitti on nullid
    tulemus = kood[0] + tulemus
    kustutada ajut 8 vasakpoolsemat bitti
```

58-ndsüsteemi kasutatakse selleks, et saada võimalikult lühikesed aadressid, millel on järgmised omadused:

- koosnevad ainult numbritest ja (inglise) tähtedest (kasulik selleks, et meilikliendid ja veebilehitsejad aadresse kahe rea vahel ei poolitaks ja hiire topeltklõps meilis või veebilehel valiks “sõnana” blokiks terve aadressi);
- ei sisalda märke, mis võivad kergesti omavahel segi minna (number ‘0’ ja suurtäht ‘O’, suurtäht ‘I’ ja väiketäht ‘l’).

Kirjutada programm, mis saab tekstina kodeeritud aadressi ja väljastab selle bitijadana.

**Sisend.** Tekstifaili `bcsis.txt` ainsal real on tekstina kodeeritud Bitcoin rahakoti aadress: numbritest ja tähtedest koosnev sõne, mille pikkus ei ületa 30 märki.

**Väljund.** Tekstifaili `bcval.txt` väljastada sisendis antud aadress jadana, milles on täpselt 168 numbrit 0 ja 1. Numbreid võib grupeerida tühikute abil ja vastuse võib jagada kuni 25 reale. Tühikuid ja reavahetusi võib panna ükskõik milliste märgipaaride vahele.

**Näide.**

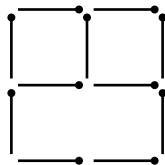
```
bcsis.txt
ComfC5sWQ6R5QZ1CtATWsr7uYhij9

bcval.txt
11000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
```

### 3. Tikuruudud

1 sekund 40 punkti

Kirjutada programm, mis saab tikkude paigutuse laual ja loendab, mitu ruutu need tikud kokku moodustavad. Näiteks alloleval joonisel kujutatud tikud moodustavad kokku 3 ruutu (kaks ruutu suurusega  $1 \times 1$  ja ühe ruudu suurusega  $2 \times 2$ ).



**Sisend.** Tekstifaili `trsis.txt` esimesel real on tikkude arv  $N$  ( $1 \leq N \leq 1000$ ). Järgmisel  $N$  real on igaühel ühe tiku asend kujul  $X Y S$ , kus  $X$  ja  $Y$  on tiku väävlita otsa täisarvulised koordinaadid ( $0 \leq X \leq 1000$ ,  $0 \leq Y \leq 1000$ ) ja  $S$  selle suund. Suund võib olla kas `H`, mis tähendab, et tiku väävliga ots on suunatud paremale, või `V`, mis tähendab, et väävliga ots on suunatud üles. Võib eeldada, et joonisel pole kuskil mitu tikku üksteise peal.

**Väljund.** Tekstifaili `trval.txt` ainsale reale väljastada moodustuvate ruutude arv.

<b>Näide.</b>	<code>trsis.txt</code>	<code>trval.txt</code>
	11	3
	0 0 H	
	1 0 H	
	0 0 V	
	2 0 V	
	0 1 H	
	1 1 H	
	0 1 V	
	1 1 V	
	2 1 V	
	0 2 H	
	1 2 H	

## 1. Бытовые приборы

1 секунда

30 очков

В одном доме необходимо в промежуток времени 00:00–24:00 выполнить  $N$  различных работ. Для выполнения каждой работы нужно использовать определённый бытовой прибор с определённой мощностью в течение заданного времени. Несколько работ можно выполнять параллельно, но каждую начатую работу нужно закончить, не прерывая её.

Написать программу, которая получает данные о цене электроэнергии за каждый час в течение суток, и находит самый дешёвый способ осуществить все работы.

**Входные данные.** На первой строке текстового файла `emsis.txt` дано число работ  $N$  ( $1 \leq N \leq 100$ ). На следующих  $N$  строках записаны данные каждой работы. На каждой строке даны три числа: количество часов (целое число  $0 \dots 3$ ) и минут (целое число  $0 \dots 59$ ), требуемые для выполнения работы, а также мощность используемой машины в киловаттах (положительное действительное число не больше 10). На последней строке файла записано 24 действительных числа: цена одного киловатт-часа для промежутков времени 00:00–01:00, 01:00–02:00, ..., 23:00–24:00 (все значения положительные и не больше 10).

**Выходные данные.** На единственной строке текстового файла `emval.txt` нужно вывести одно действительное число — наименьшую возможную суммарную стоимость энергии, требуемой для выполнения всех работ к полуночи. Абсолютная погрешность выведенного ответа не должна превышать 0.01.

**Пример.**

<code>emsis.txt</code>	<code>emval.txt</code>
2	15.00
1 0 3.0	
2 0 1.5	
5 5 5 5 2 5 3 3 5 5 5 5 5 5 5 5 5 5 5 5 5 5	

Для выполнения первой работы необходимо использовать машину мощностью 3 кВт в течение ровно часа. Это выгоднее всего делать в промежуток времени 04:00–05:00, когда цена за киловатт-час равна 2.00, и стоимость энергии на выполнение этой работы, таким образом, равна 6.00.

Для выполнения второй работы необходимо использовать машину мощностью 1.5 кВт в течение двух часов. Это выгоднее всего делать в промежуток времени 06:00–08:00, когда цена за киловатт-час равна 3.00. Суммарная стоимость энергии на выполнение этой работы в это время будет 9.00.

## 2. Bitcoin

1 секунда

30 очков

Bitcoin — это криптографическая денежная система. В этой системе виртуальные монеты хранятся в виртуальных “кошельках”. У каждого кошелька есть уникальный адрес, который является 168-битным целым числом.

Для записи адреса в текстовом виде, его преобразуют в 58-чную систему с помощью следующего алгоритма (здесь `address` — переменная, в которой задан адрес в виде большого целого числа; `tulemus` — переменная, в которой в конце оказывается представление адреса в текстовом виде; `div` обозначает целочисленное деление; `mod` — нахождение остатка):

```
tulemus = ""
kood[0..57] = "123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
ajut = address
while ajut > 0
    jaak = ajut mod 58
    ajut = ajut div 58
    tulemus = kood[jaak] + tulemus
ajut = address
while ajut 8 vasakpoolsemat bitti on nullid
    tulemus = kood[0] + tulemus
    kustutada ajut 8 vasakpoolsemat bitti
```

58-чная система была выбрана для кодирования адресов, потому что таким образом можно получить относительно короткую запись адреса, которая удовлетворяет следующим свойствам:

- адреса состоят только из цифр и латинских букв (это удобно при выводе адресов в браузере или почтовых клиентах).
- адреса не содержат знаков, которые легко перепутать (цифра ‘0’ и буква ‘O’, заглавная буква ‘I’ и прописная ‘l’).

Написать программу, которая получает закодированный вышеописанным способом в текстовом виде адрес, и выводит адрес в виде последовательности битов.

**Входные данные.** На единственной строке текстового файла `bcsis.txt` записан адрес кошелька Bitcoin — строка, состоящая из цифр и букв, длина которой не превышает 30 символов.

**Выходные данные.** В текстовый файл `bcval.txt` вывести данный адрес в виде двоичной последовательности, в которой ровно 168 знаков 0 или 1. Цифры можно (но не обязательно) группировать произвольным образом с помощью пробелов или переносов строки (однако всего в файле не может быть более 25 строк).

**Пример.**

```
bcsis.txt
ComfC5sWQ6R5QZ1CtATWsr7uYhij9

bcval.txt
11000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000
```

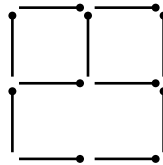


### 3. Квадраты из спичек

1 секунда

40 очков

Написать программу, которая получает расположение спичек на столе, и определяет, сколько всего квадратов содержит данная конфигурация спичек. Например, фигура из спичек на следующем рисунке содержит 3 квадрата (два квадрата размером  $1 \times 1$  и один размером  $2 \times 2$ ).



**Входные данные.** На первой строке текстового файла `trsis.txt` дано количество спичек  $N$  ( $1 \leq N \leq 1000$ ). На каждой из следующих  $N$  строк дано расположение одной из спичек в виде  $X Y S$ , где  $X$  и  $Y$  — целочисленные координаты “чистого” конца спички ( $0 \leq X \leq 1000$ ,  $0 \leq Y \leq 1000$ ), а  $S$  — направление. Направление может быть либо **H**, что означает, что другой конец спички направлен вправо, или **V**, что означает, что другой конец спички направлен вверх. Можно предполагать, что никакие две спички не лежат одна на другой.

**Выходные данные.** На единственной строке текстового файла `trval.txt` вывести количество квадратов, присутствующих в описанном расположении спичек.

Пример.	<code>trsis.txt</code>	<code>trval.txt</code>
	11	3
	0 0 H	
	1 0 H	
	0 0 V	
	2 0 V	
	0 1 H	
	1 1 H	
	0 1 V	
	1 1 V	
	2 1 V	
	0 2 H	
	1 2 H	

## 1. Elektrimasinad

1 sekund 20 punkti

Ühes kodus on vaja ajavahemikul kell 00:00–24:00 ära teha  $N$  majapidamistööd. Iga töö jaoks on kindla võimsusega kodumasin ja töö tegemiseks kulub kindel aeg. Mitut tööd võib teha ka samaaegselt, kuid iga alustatud töö tuleb ära teha ilma seda vahepeal katkestamata.

Kirjutada programm, mis saab elektrienergia hinna ööpäeva igas tunnis ja leiab minimaalse kulu, millega on võimalik kõik tööd tehtud saada.

**Sisend.** Tekstifaili `emsis.txt` esimesel real on tööde arv  $N$  ( $1 \leq N \leq 100$ ). Järgmisel  $N$  real on tööde andmed. Igal real on töö tegemiseks kuluv aeg tundides (täisarv  $0 \dots 3$ ) ja minutites (täisarv  $0 \dots 59$ ) ning masina võimsus kilovattides (positiivne reaalarv, mis ei ületa 10). Faili viimasel real on 24 reaalarvu: ühe kilovatt-tunni elektrienergia hind kell 00:00–01:00, 01:00–02:00, ..., 23:00–24:00 (positiivsed reaalarvud, mis ei ületa 10).

**Väljund.** Tekstifaili `emval.txt` ainsale reale väljastada üks reaalarv: minimaalne rahasumma, mis tuleb elektrienergiale kulutada, et kõik tööd südaööks tehtud saada. Vastus ei tohi täpselt väärtusest erineda rohkem kui 0,01 võrra.

**Näide.**

<code>emsis.txt</code>	<code>emval.txt</code>
2	15.00
1 0 3.0	
2 0 1.5	
5 5 5 5 2 5 3 3 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5	

Esimese töö tegemiseks kulub 3 kW võimsusega masinal täpselt üks tund. Seda on kõige odavam teha kell 04:00–05:00, mil kWh hind on 2.00 ja töö tegemiseks kulub 6.00 väärtuses energiat.

Teise töö tegemiseks kulub 1,5 kW võimsusega masinal täpselt kaks tundi. Seda on kõige odavam teha kell 06:00–08:00, mil kWh hind on 3.00 ja töö tegemiseks kulub 9.00 väärtuses energiat.

## 2. Kaardilugemine

1 sekund 40 punkti

$N \times M$  kuusnurksest pikslit koosneval kaardil on naaberriigid tähistatud erinevate värvidega, kuid riigid, mis ei ole kõrvuti, võivad olla ka sama värvi. Täpsemalt kuuluvad kaks ühise servaga pikslit samasse riiki parajasti siis, kui nad on sama värvi.

Lisaks on teada, et kaardil pole mitmest eraldiseisvast osast koosnevaid riike.

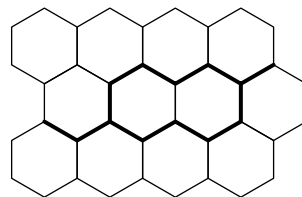
Kirjutada programm, mis leiab kaardil olevate riikide arvu ja riigipiiride kogupikkuse.

**Sisend.** Tekstifaili `kxsis.txt` esimesel real on kaardi ridade arv  $N$  ( $1 \leq N \leq 300$ ) ja veergude arv  $M$  ( $1 \leq M \leq 300$ ). Järgmisel  $N$  real on igaühel  $M$  tühikutega eraldatud täisarvu  $K_{ij}$  ( $0 \leq K_{ij} \leq 100$ ): pikslite värvikoodid ridade kaupa. Pikslid on alati paigutatud nii, nagu näha alloleval joonisel: teise rea pikslid on esimese rea omadest veidi paremal, kolmanda rea omad teise rea omadest veidi vasakul j.n.e.

**Väljund.** Tekstifaili `kxval.txt` esimesele reale väljastada kaardil olevate riikide arv ja teisele reale riigipiiride kogupikkus. Riigipiiridena arvestada ainult kaardil nähtavaid riikidevahelisi jooni, kaardi servi piirideks mitte lugeda.

<b>Näide.</b>	<code>kxsis.txt</code>	<code>kxval.txt</code>
	3 4	3
	1 1 1 1	13
	1 2 2 3	
	3 3 3 3	

Kaardil näha olevad riigipiirid on alloleval joonisel tähistatud rasvasema joonega.



<b>Näide.</b>	<code>kxsis.txt</code>	<code>kxval.txt</code>
	3 4	3
	1 2 2 1	10
	1 2 2 1	
	1 2 2 1	

**Hindamine.** Selles ülesandes hinnatakse eraldi riikide arvu ja piiride pikkuse leidmist. Programm, mis ei leia riikide arvu, peab faili esimesele reale väljastama arvu  $-1$ . Programm, mis ei leia piiride pikkust, peab faili teisele reale väljastama arvu  $-1$ . Ainult ühe arvu väljastamine on viga ja sellise väljundi eest punkte ei anta.

### 3. Paarikaupa lahutamine

3 sekundit 40 punkti

$N \times M$  tabeli igas lahtris on täisarv. Edasi võib igal sammul valida kaks lahtrit, millel on ühine serv, ja vähendada mõlemas lahtris olevaid arve 1 võrra.

Kirjutada programm, mis leiab, kas selliste sammudega on võimalik saavutada olukord, kus tabeli kõigis lahtrites on nullid.

**Sisend.** Tekstifaili `plsis.txt` esimesel real on tabeli ridade arv  $N$  ( $1 \leq N \leq 10$ ) ja veergude arv  $M$  ( $1 \leq M \leq 10$ ). Järgmisel  $N$  real on igalühel  $M$  tühikutega eraldatud mittenegatiivset täisarvu: tabeli elemendid. On teada, et tabeli kõigi elementide summa ei ületa 120. Tabeli read on nummerdatud  $1 \dots N$  ülalt alla ja veerud  $1 \dots M$  vasakult paremale.

**Väljund.** Kui kõigi lahtrite nullimine pole võimalik, väljastada tekstifaili `plval.txt` ainsale reale tekst `EI SAA`. Kui nullimine on võimalik, väljastada faili esimesele reale tekst `SAAB` ja järgmistele ridadele selleks vajalikud sammud. Iga sammu kirjeldus väljastada eraldi reale kujul  $R_1 V_1 R_2 V_2$ , kus  $(R_1, V_1)$  on ühe ja  $(R_2, V_2)$  teise sellel sammul valitud lahtri rea- ja veerunumber. Kui võimalikke sammujadasid on mitu, väljastada ükskõik milline neist.

<b>Näide.</b>	<code>plsis.txt</code>	<code>plval.txt</code>
	2 2	SAAB
	2 1	1 1 1 2
	2 1	1 1 2 1
		2 1 2 2

<b>Näide.</b>	<code>plsis.txt</code>	<code>plval.txt</code>
	1 2	EI SAA
	0 3	

**Hindamine.** Selles ülesandes antakse `EI SAA` vastusega testide eest punkte ainult neile programmidele, mis lahendavad õigesti vähemalt ühe testi, kus kogu tabeli nullimine on võimalik.

## 1. Бытовые приборы

1 секунда

20 очков

В одном доме необходимо в промежуток времени 00:00–24:00 выполнить  $N$  различных работ. Для выполнения каждой работы нужно использовать определённый бытовой прибор с определённой мощностью в течение заданного времени. Несколько работ можно выполнять параллельно, но каждую начатую работу нужно закончить, не прерывая её.

Написать программу, которая получает данные о цене электроэнергии за каждый час в течение суток, и находит самый дешёвый способ осуществить все работы.

**Входные данные.** На первой строке текстового файла `emsis.txt` дано число работ  $N$  ( $1 \leq N \leq 100$ ). На следующих  $N$  строках записаны данные каждой работы. На каждой строке даны три числа: количество часов (целое число  $0 \dots 3$ ) и минут (целое число  $0 \dots 59$ ), требуемые для выполнения работы, а также мощность используемой машины в киловаттах (положительное действительное число не больше 10). На последней строке файла записано 24 действительных числа: цена одного киловатт-часа для промежутков времени 00:00–01:00, 01:00–02:00, ..., 23:00–24:00 (все значения положительные и не больше 10).

**Выходные данные.** На единственной строке текстового файла `emval.txt` нужно вывести одно действительное число — наименьшую возможную суммарную стоимость энергии, требуемой для выполнения всех работ к полуночи. Абсолютная погрешность выведенного ответа не должна превышать 0.01.

<b>Пример.</b>	<code>emsis.txt</code>	<code>emval.txt</code>
	2	15.00
	1 0 3.0	
	2 0 1.5	
	5 5 5 5 2 5 3 3 5 5 5 5 5 5 5 5 5 5 5 5 5	

Для выполнения первой работы необходимо использовать машину мощностью 3 кВт в течение ровно часа. Это выгоднее всего делать в промежуток времени 04:00–05:00, когда цена за киловатт-час равна 2.00, и стоимость энергии на выполнение этой работы, таким образом, равна 6.00.

Для выполнения второй работы необходимо использовать машину мощностью 1.5 кВт в течение двух часов. Это выгоднее всего делать в промежуток времени 06:00–08:00, когда цена за киловатт-час равна 3.00. Суммарная стоимость энергии на выполнение этой работы в это время будет 9.00.

## 2. Картография

1 секунда 40 очков

Имеется карта, состоящая из  $N \times M$  шестиугольных пикселей. Раскраска пикселей на карте обозначает различные страны. Любые две соседние страны раскрашены в разные цвета. При этом на карте могут быть разные, не находящиеся рядом страны, для которых используется один и тот же цвет пикселей. Точнее говоря, два соседних пикселя принадлежат одной и той же стране тогда и только тогда, когда они раскрашены в один и тот же цвет.

На карте нет стран, территория которых состояла бы из нескольких несвязанных частей.

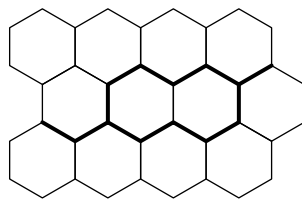
Написать программу, которая считала бы число стран на карте и суммарную длину границ между ними.

**Входные данные.** На первой строке текстового файла `kxsis.txt` даны размеры карты: количество строк  $N$  ( $1 \leq N \leq 300$ ) и столбцов  $M$  ( $1 \leq M \leq 300$ ). На каждой из следующих  $N$  строк записаны  $M$  разделённых пробелами целых числа  $K_{ij}$  ( $0 \leq K_{ij} \leq 100$ ): цвета пикселей соответствующего ряда (каждое число соответствует какому-то цвету, разные числа соответствуют разным цветам). Пиксели всегда расположены так, как указано на нижеприведённом рисунке: пиксели второй строки сдвинуты относительно первой слегка вправо, пиксели третьей сдвинуты относительно второй строки слегка влево, итд.

**Выходные данные.** На первой строке текстового файла `kxval.txt` вывести число стран на карте. На второй строке — суммарную длину межгосударственных границ. В качестве границ нужно считать только линии, проходящие между странами — края карты за границу не считаются.

<b>Пример.</b>	<code>kxsis.txt</code>	<code>kxval.txt</code>
	3 4	3
	1 1 1 1	13
	1 2 2 3	
	3 3 3 3	

Границы стран отмечены на нижеприведённом рисунке жирной линией.



<b>Пример.</b>	<code>kxsis.txt</code>	<code>kxval.txt</code>
	3 4	3
	1 2 2 1	10
	1 2 2 1	
	1 2 2 1	

**Оценивание.** В этом задании задачи нахождения числа стран и суммарной длины границ оцениваются отдельно. Программа, которая не может определить число стран, должна вывести на первой строке выходного файла число  $-1$ . Программа, которая не может определить суммарную длину границ должна вывести на второй строке файла число  $-1$ . Программа, которая выводит только одно число считается ошибочной, и за такое решение очки выдаваться не будут.

### 3. Попарное вычитание

3 секунды

40 очков

В каждой ячейке таблицы  $N \times M$  записано целое число. Каждым шагом можно выбрать две расположенные рядом по горизонтали или вертикали ячейки, и уменьшить оба числа на единицу.

Написать программу, которая определяет, можно ли с помощью таких операций достичь ситуации, когда во всех ячейках таблицы нули.

**Входные данные.** На первой строке текстового файла `plsis.txt` дано количество строк  $N$  ( $1 \leq N \leq 10$ ) и столбцов  $M$  ( $1 \leq M \leq 10$ ) таблицы. На каждой из следующих  $N$  строк записано  $M$  разделённых пробелами неотрицательных целых чисел: элементы таблицы. Известно, что сумма всех элементов не превышает 120. Строки таблицы пронумерованы  $1 \dots N$  сверху вниз, и столбцы  $1 \dots M$  слева направо.

**Выходные данные.** Если обнулить все ячейки одновременно невозможно, вывести на единственной строке текстового файла `plval.txt` текст `EI SAA`. Если ячейки можно обнулить, вывести на первой строке файла текст `SAAB`, а на следующих строках необходимые для этого шаги. Каждый шаг вывести на отдельной строке в виде  $R_1 V_1 R_2 V_2$ , где  $(R_1, V_1)$  — это номер строки и столбца первой, а  $(R_2, V_2)$  — номер строки и столбца второй выбранной на этом шагу ячейки. Если возможных решений несколько, вывести любое.

<b>Пример.</b>	<code>plsis.txt</code>	<code>plval.txt</code>
	2 2	SAAB
	2 1	1 1 1 2
	2 1	1 1 2 1
		2 1 2 2

<b>Пример.</b>	<code>plsis.txt</code>	<code>plval.txt</code>
	1 2	EI SAA
	0 3	

**Оценивание.** В этой задаче за тесты с ответом `EI SAA` очки выдаются только тем программам, которые корректно решат хотя бы один тест, в котором обнуление таблицы возможно.