

ПРОЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Плодив, 14 юни 2008 г.

Група А (11-12 клас)

Задача А1. КВАДРАТИ

В равнината са дадени са n квадрата: S_1, S_2, \dots, S_n .

Разглеждаме неориентиран граф с върхове дадените квадрати, като между два квадрата има ребро, тогава и само тогава, когато двата квадрата имат общи вътрешни точки.

Напишете програма **square**, която намира броя на ребрата в най-късия път от S_1 до S_n .

На първия ред на стандартния вход е записано числото n ($2 \leq n \leq 1000$), а на всеки от следващите n реда са записани по четири цели числа x_M, y_M, x_A, y_A , задаващи координатите на центъра (x_M, y_M) и на един от върховете (x_A, y_A) на поредния квадрат. Всички координати са в интервала $[-10000, +10000]$. Номерацията на квадратите е според появяването им във входния файл.

На стандартния изход да се изведе търсеният брой или -1 , ако не съществува път от S_1 до S_n .

ПРИМЕР

Вход

```
3
0 0 1 1
1 0 2 1
4 0 7 2
```

Изход

```
2
```

Задача A2. ЕГИПЕТСКИ СУМИ

Древните египтяни са използвали само дробни с числител 1. Всяко друго рационално число те записвали като сума от дробни с числител 1. Например $\frac{7}{9} = \frac{1}{2} + \frac{1}{4} + \frac{1}{36}$.

. Нека $a_1 \leq a_2 \leq \dots \leq a_n$ са n на брой естествени числа, за които $\frac{1}{a_1} + \frac{1}{a_2} + \dots + \frac{1}{a_n} = 1$.

Напишете програма **egypt**, която по зададена стойност на n намира най-голямата възможна стойност на a_n .

От един ред на стандартния вход се въвежда n ($2 \leq n \leq 20$).

На един ред на стандартния изход програмата трябва да изведе едно цяло число, равно на търсената най-голяма стойност на a_n .

ПРИМЕР

Вход

3

Изход

6

Задача А3. ЩАСТЛИВИ ЧИСЛА

Някои естествени числа имат в десетичния си запис забележителни свойства. Ще наречем „щастливи“ такива числа, произведението от десетичните цифри на които се дели на сумата им без остатък. Интересен пример е числото 36. Произведението от цифрите му е 18, а сумата им е 9. Тъй като 18 се дели на 9 без остатък, 36 е щастливо число. За едноцифрените числа (нулата не причисляваме към естествените числа) ще считаме, че „сумата“ и „произведението“ на цифрите им е самото число, тъй че, съгласно определението, те също са щастливи. Напишете програма **lucky**, която за въведен затворен интервал от естествени числа $[a, b]$ определя броя на щастливите числа в него.

От стандартния вход се въвежда един ред с двете естествени числа a и b , разделени с интервал, като $a \leq b$. Никое от числата няма повече от 14 десетични цифри.

Запишете на стандартния изход един ред с броя на щастливите естествени числа в интервала $[a, b]$.

ПРИМЕР

Вход

5 73

Изход

17

Пояснение: Щастливите числа в интервала $[5, 73]$ са 5, 6, 7, 8, 9, 10, 20, 22, 30, 36, 40, 44, 50, 60, 63, 66 и 70.

ПРОЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Плодив, 15 юни 2008 г.

Контролно състезание

Задача А4. РАЗБИВАНЕ

Дадено е множество от n естествени чила ($0 < n < 25$), всяко със стойност в интервала от 1 до 50 000. Напишете програма **part**, която определя дали числата от даденото множество могат да бъдат разпределени в три непересичащи се подмножества така, че сумата от елементите в трите подмножества да е еднаква и обединението на трите подмножества да съвпада с даденото множество.

Програмата прочита от стандартния вход стойността на n , следвана от стойностите на числата от даденото множество. Измежду тези стойности няма постарящи се. Всички числа във входа са цели, положителни и са разделени с по един интервал.

На стандартния изход програмата трябва да изведе цифрата 1, ако описаното разбиване е възможно, или 0, ако не е възможно.

Забележка: Оценяването на програмата ще бъде извършено като тестовите за проверката ще бъдат групирани в 10 групи, всяка съставена от 2 или повече теста. По 10 т. ще бъдат присъждани за група тестове, само ако програмата е дала верен отговор за всичките тестове от групата.

ПРИМЕР 1

Вход

6

1 8 11 3 4 6

Изход

1

ПРИМЕР 2

Вход

5

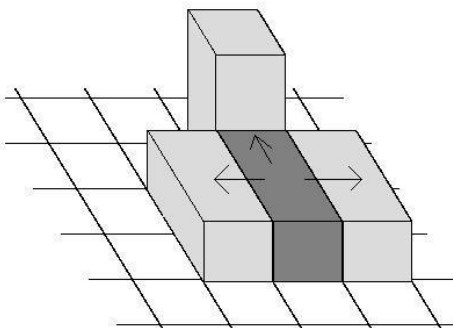
8 11 3 4 6

Изход

0

Задача A5. ИГРА

Дадено е игрално поле с размери N на M ($1 \leq N, M \leq 20$), съставено от единични квадрати. Всеки квадрат може да е или плочка, или дупка. Някъде върху полето е поставено, в изправено положение, трупче с размери 2 на 1 на 1, което играчът мести. Той може да прави един от четири хода – да преобръща трупчето по някой от ръбовете на основата му (основата е страната, на която лежи).



Пример за 3 от 4-те възможни хода, когато трупчето е легнало

Ако някоя част от трупчето след ход се окаже над дупка, то пада от дъската и играта завършва. Дадено е, че едно от квадратчетата на игралното поле е „изход” когато постави трупчето в изправено положение върху това поле, играчът печели.

По полето са разположени K ключа и P подвижни плочки (врати) ($0 \leq K \leq 20$, $0 \leq P \leq 10$). Тези плочки могат да бъдат активирани (по тях може да се минава) или дезактивирани (все едно на полето има дупка). Ключовете се активират когато поне една от двете части на трупчето е върху тях. Един ключ може да отваря, да затваря или да сменя състоянието на една или повече подвижни плочки. Това става след края на хода.

Освен това, някои плочки са чупливи. Върху тях трупчето може да бъде поставено, но само в легнало положение. Т. е, ако се окаже изправено върху чуплива плочка, тя не издържа на теглото и пропада. Напишете програма **bloxorz**, която намира дали, и по какъв начин може да се достигне до изхода.

Вход

Ред 0: две числа, N и M

Ред 1, ..., N : по M символа, всеки от които е един от следните: '.' – плочка, '_' – дупка, '^' – чуплива плочка, '0' – врата с номер 0, '1' – врата с номер 1, ..., '9' – врата с номер 9, 'S' – начална позиция на трупчето, 'E' – изход, 'K' – ключ

Ред $N + 1$, ..., $N + K$: по един ред за всеки ключ, като първи са по-горните ключове, а при равна височина – тези, които са по-наляво. Редове са съставени от последователност от двойки символи $X\#$, където X е от множеството {'O', 'C', 'T'} (отваря, затваря, променя врата), а $\#$ е число от 0 до 9, показващо за коя врата се отнася командата.

Ред $N + K + 1$: един ред от символи '0' и '1', като i -тият символ показва началното състояние на врата i (0 – отворена/дупка, 1 – по нея може да се минава)

Изход

Ред 1: Минималният брой ходове, с които може да се достигне до изхода, или 0, ако това не е възможно.

Ред 2: записваме ако първият ред не е "0" – самите ходове, като последователност от символи (u/d/l/r за нагоре/надолу/наляво/надясно), като j -тият от тях показва в каква посока е обърнато трупчето на j -тия ход.

ПРИМЕР

Вход

```
4 6
S..._
...0
...1
K...E
00 01
00
```

Изход

```
12
rrddllrrurrd
```

Забележка: 40% от тестовете са без врати/ключове.

Уточнения: Всички ключове, началната и крайната позиции са върху нечупливи, неподвижни плочки. Част от трупчето може да застава върху изхода в легнало положение, но тогава играта не завършва. Няма случай, в който в един и същи ход една врата едновременно да трябва да се отвори и да се затвори.

Задача А6. АЛЯСКА

По време на златната треска в Аляска двама златотърсачи се заели да заграждат участъци за добив. Единият от тях забил на едно място кол за ограда, повървял по права линия, забил нов кол, огледал се, избрал нова посока, повървял по нея и т.н., докато стигнал в изходната точка. Докато обикалял, внимавал да избира местата за колове така, че бъдещата ограда да не се пресича. Другият постъпил по подобен начин, само че вратът му бил схванат, поради което той се оглеждал и завивал винаги в една и съща посока – или все наляво, или все надясно.

Напишете програма **alaska**, която определя дали двата заградени участъка имат обща площ и ако е така, намира лицето ѝ.

От стандартния вход се четат редове с двойки цели числа във всеки – координатите на коловете за ограда на първия златотърсач. Следва празен ред, а след него – координатите на коловете за ограда на втория златотърсач в същия вид като за първия. Коловете се дават в реда, в който са поставени.

На стандартния изход се извежда един ред, в който стои число с два знака след десетичната точка – приблизителна стойност на лицето на общата за двата участъка площ – или знакът 0 (нула), ако такава площ няма. Изведеното число трябва да бъде възможно най-близко до действителното лице.

Ограничения

Броят колове за всяка от оградите е не по-голям от 1000.

ПРИМЕР

Вход	Изход
5 0	12.94
10 5	
5 10	
0 5	
0 0	
5 5	
1 10	