НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА
Общински кръг, януари 2021 г.
Група В, 9-10 клас

Задача В1. Уравнение

Дадени са целите положителни числа a, b и c, и цялото число p. Напишете програма **equation**, която намира две цели числа x и y, за които е изпълнено, че $a \cdot x + b \cdot y = c$ и освен това, стойността на x да е равна на p, а ако няма такава, тогава x да е възможно най-близко до даденото цяло число p, т.е *абсолютната стойност* $|x - p|$ *да е най-малка*. Когато има повече от една такава стойност за x, програмата трябва да изведе най-голямата.

Вход. Вашата програма трябва да прочете от стандартния вход броя N на тестовете. Всеки тест е зададен на отделен ред с четири цели числа a, b, c и p, които са разделени с интервали.

Изход. Вашата програма трябва да изведе на отделни редове в стандартния изход, според последователността на тестовете от входа, по две цели числа, разделени с точно един интервал: търсената стойност на x и съответната стойност на y. Когато не съществува двойка x и y, удовлетворяващи условието на задачата, Вашата програма трябва да изведе две нули, разделени с един интервал.

Ограничения: $0 < N < 100$; $0 < a,b,c < 10^9$; $-10^9 < p < 10^9$.

Пример

Вход

```
3
7 6 5 8
2 4 1 1
2 3 1 -4
```

Изход

```
11 -12
0 0
-4 3
```

НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА
Общински кръг, януари 2021 г.
Група В, 9-10 клас

Задача В2. Матрици

Нека **A** е матрица с размерност $m \times k$ (m реда и k колони), а **B** е матрица с размерност $k \times p$ (k реда и p колони). Елементите на **A** ще означаваме с a_{ij}, където i е номерът на реда ($1 \le i \le m$), а j е номерът на колоната ($1 \le j \le k$). Съответно елементите на **B** ще означаваме с b_{ij}. Забележете, че броят на колоните на **A** е равен на броя на редовете на **B.** Предполагаме, че елементите и на двете матрици са реални числа.

За такива две матрици се дефинира операцията произведение, като резултатът е матрица **C = A × B** с m реда и p колони, в която $c_{ij} = a_{i1}*b_{1j} + a_{i2}*b_{2j} + \ldots\ldots\ a_{ik}*b_{kj}$ ($1 \le i \le m$, $1 \le j \le p$). За изчисляването на всеки елемент на матрицата **C** се извършват k умножения на реални числа, което значи, че за получаването на цялата матрица **C** ще се извършат $m*p*k$ умножения на реални числа.

Ако са налице три матрици **A**, **B** и **C** с размерности $m \times k$, $k \times p$ и $p \times q$, то лесно се вижда, че **(A × B) × C = A × (B × C)**, което ни позволява да дефинираме произведението **A × B × C,** което ще бъде матрица с размерност $m \times q$, като резултатът не зависи от това дали първо ще умножим **A** по **B** и след това получената матрица ще умножим по **C** или първо ще умножим **B** по **C** и след това **A** ще умножим по получената матрица. Това свойство на умножението на матрици се нарича **асоциативност.**

По този начин можем да дефинираме произведение на произволен брой матрици $\mathbf{M_1} \times \mathbf{M_2} \times \ldots\ldots \times \mathbf{M_n}$ с размерности $R_0 \times R_1$, $R_1 \times R_2$, $R_2 \times R_3\ldots\ldots$, $R_{n-1} \times R_n$ като свойството асоциативност ни гарантира, че резултатът ще бъде един и същ, независимо в какъв ред извършваме умножението.

Резултатът ще бъде един и същ, но броят на умноженията на реалните числа за получаването на този резултат може съществено да се различава. Да разгледаме следния пример:

Трябва да се намери произведението $\mathbf{M_1} \times \mathbf{M_2} \times \mathbf{M_3} \times \mathbf{M_4}$, където матриците $\mathbf{M_1}$, $\mathbf{M_2}$, $\mathbf{M_3}$ и $\mathbf{M_4}$ са съответно с размерности 10 × 20, 20 × 50, 50 × 1 и 1 × 100. Ако пресмятаме призведението в реда, определен от скоби, по следния начин $\mathbf{M_1 \times (M_2 \times (M_3 \times M_4))}$, то ще получим следния брой умножения на реални числа:

50*1*100 = 5000 - от $\mathbf{(M_3 \times M_4)}$ + (получава се матрица с размерност 50 × 100)
20*50*100 = 100000 – от $\mathbf{M_2 \times (M_3 \times M_4)}$ + (получава се матрица с размерност 20 × 100)
10*20*100 = 20000 – от $\mathbf{M_1 \times (M_2 \times (M_3 \times M_4))}$ (получава се матрица 10 × 100)

При тази последователност на умножаване на матриците са нужни 125 000 умножения на реални числа. Нека да извършим умножението в друг ред: $\mathbf{(M_1 \times (M_2 \times M_3)) \times M_4}$. Броят умножения на реални числа за получаване на същия резултат ще бъде:

20*50*1 = 1000 – от $\mathbf{(M_2 \times M_3)}$ + (получава се матрица с размерност 20 × 1)
10*20*1 = 200 – от $\mathbf{(M_1 \times (M_2 \times M_3))}$ + (получава се матрица с размерност 10 × 1)
10*1*100 = 1000 – от $\mathbf{(M_1 \times (M_2 \times M_3)) \times M_4}$ (получава се матрица с размерност 10 × 100).

При тази последователност на умножаване на матриците са нужни 2200 умножения на реални числа – разликата е чувствителна, нали?

Напишете програма **matrices,** която намира минималния брой умножения на реални числа, които са необходими, за да бъде намерено произведението на n матрици $\mathbf{M_1} \times \mathbf{M_2} \times \ldots\ldots \times \mathbf{M_n}$ с размерности $R_0 \times R_1$, $R_1 \times R_2$, $R_2 \times R_3 \ldots\ldots$, $R_{n-1} \times R_n$.

Вход. От първия ред на стандартния вход се въвежда цяло положително число n – брой на матриците. От втория ред се въвеждат $n+1$ цели положителни числа $R_0, R_1, R_2 \ldots\ldots, R_n$, разделени с интервали, които задават размерностите на матриците.

Изход. На един ред на стандартния изход изведете само едно число – намерения минимален брой умножения на реални числа, необходими за намирането произведението на матриците.

Ограничения: $2 \leq n \leq 100$;

$1 \leq R_0, R_1, R_2 \ldots\ldots, R_n \leq 1000$

Пример (съответства на примера, разгледан в условието)

Вход

```
4
10 20 50 1 100
```

Изход

```
2200
```

НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА
Общински кръг, януари 2021 г.
Група B, 9-10 клас

Задача B3. xor0

Любимата операция на Дени е побитово *изключващо или* (ще я означаваме с ^). Логическото *изключващото или* има следната истинна таблица:

a	b	a ^ b
0	0	0
0	1	1
1	0	1
1	1	0

Съответно побитовото *изключващо или* приложено на две числа се пресмята като се приложи логическото *изключващо или* на съответстващите битове на двете числа. Например ако $x = 5 = 101_{(2)}$ и $y = 11 = 1011_{(2)}$, то:

$x = 5 =$	0	1	0	1
$y = 11 =$	1	0	1	1
x^$y = 14 =$	1	1	1	0

Дени разполага с масив a от $\boldsymbol{N}$ неотрицателни цели числа и започва да прилага любимата си операция над произволни негови подмасиви (подмасив е всяка редица от последователни елементи на масива). Така например за подмасив $a[l]$, $a[l+1]$, ..., $a[r]$ тя намира $a[l]$^$a[l+1]$^...^$a[r]$. Тази стойност тя нарича *xor* на подмасива. Оказало се, че доста често получава 0. Затова тя се чуди колко са подмасивите с *xor* равен на 0. Сега идва Вашият ред – напишете програма **xor0**, която намира тази бройка за Дени.

Вход

От първия ред на стандартния вход се въвежда числото $\boldsymbol{N}$ – броят на числата в масива a. От следващия ред се въвеждат $\boldsymbol{N}$ неотрицателни цели числа – числата на масива.

Изход

Единствено число – броят подмасиви с *xor* равен на 0.

Ограничения

- $1 \leq \boldsymbol{N} \leq 5.10^5$
- $0 \leq a[i] \leq 10^9$
- в 50% от тестовете: $\boldsymbol{N} \leq 10^3$

Пример

Вход	Изход	Обяснение на примера
4 1 1 5 4	2	Подмасивите с xor равен на 0 са: 1,1 и 1, 5, 4.