

Задача А. Випробування автомата

Input file name:	testing.in
Output file name:	testing.out
Time limit:	100 ms
Memory limit:	128 M

Фірма bookface, яка створена в Ужляндії, в якій працює Степан, вирішила встановити в своїх офісах автомати з продажу чаю та кави, щоб програмісти під час перерви могли з толком провести час.

Вартість склянки чаю та кави в автоматі передбачається встановити рівній п'яти ужикам (така в Ужляндії валюта). Автомати будуть приймати монети по 5 і 10 ужиків, а також купюри в 10, 50 і 100 ужиків. Коли програмісту потрібно видавати здачу (тобто коли програміст кинув у автомат монету в 10 ужиків, або купюру в 10, 50 або 100ужиків), автомат видає здачу монетами в п'ять ужиків; якщо ж пасажир кинув у автомат монету в п'ять ужиків, то автомат її зберігає і може використовувати для здачі наступним програмістам.

Очевидно, що, щоб забезпечити можливість видачі здачі всім ппрограмістам, може знадобитися спочатку завантажити в автомат деяку кількість монет в п'ять ужиків. Зараз в офісах фірми проходять випробування з метою визначити мінімальну кількість монет, які треба завантажити в автомат перед робочим днем.

Вам дано протокол одного з таких випробувань: відомий порядок, в якому програмісти оплачували свої покупки різними монетами і купюрами. Визначте, яку мінімальну кількість монет в п'ять ужиків, повинно було спочатку перебувати в автоматі, щоб усім пасажиром вистачило здачі.

Вхідні дані: У першому рядку вхідного файлу знаходиться одне натуральне число N - кількість покупок в автоматі, які були здійснені в ході випробування ($1 \leq N \leq 50\ 000$). У другому рядку знаходяться N натуральних чисел, кожне з яких рівне номіналу монети або купюри, яку використовував черговий програміст для оплати; кожен номінал може приймати одне з чотирьох значень: 5, 10, 50 або 100.

Вихідні дані: У вихідний файл виведіть одне число - мінімальну кількість монет в п'ять ужиків, які треба було завантажити в автомат спочатку, щоб усім програмістам вистачило здачі.

Приклади вхідних та вихідних даних:

testing.in	testing.out	Пояснення
3 10 5 100	19	У першому прикладі одна монета в п'ять ужиків буде потрібна для здачі першому програмісту і 19 монет - третьому, але під час здачі третьому можна використовувати ту монету, яку кине другий програміст, тому спочатку у автоматі досить 19 монет.
3 5 5 10	0	У другому прикладі здачу третьому програмісту можна видати, використовуючи монету першого або другого покупця, і тому не потрібно завантажувати монети в автомат спочатку.
4 50 5 5 5	9	У третьому прикладі першове програмісту потрібні дев'ять монет здачі, та всі вони повинні спочатку знаходитися в автоматі.

Задача В. Кросворд

Input file name:	crossword.in
Output file name:	crossword.out
Time limit:	1200 ms
Memory limit:	128 M

Вам дано квадратний кросворд розміру $N \times N$. Порожні клітини позначені в ньому символом '-', зафарбовані - символом '#'. За правилами кросвордів, кожне слово має складатися мінімум з 2 букв. Вам потрібно для даного кросворду порахувати кількість слів по горизонталі і по вертикалі.

Вхідні дані: У першому рядку міститься число N ($1 \leq N \leq 2000$) - розмір кросворду. Наступні N рядків містять кросворд. Кожен рядок складається з N символів '-' і '#', описаних вище.

Вихідні дані: Виведіть два числа - кількість слів по горизонталі і по вертикалі.

Приклади вхідних та вихідних даних:

crossword.in	crossword.out
5 ----# --##- ----- -##-- #-----	5 4

Задача С. Цікава задача

Input file name:	interesting.in
Output file name:	interesting.out
Time limit:	200 ms
Memory limit:	128 M

Арифметична прогресія це послідовність чисел виду

$$a_1, a_1 + d, a_1 + 2d, \dots, a_1 + (n-1)d, \dots$$

де a_1 — це перший член прогресії, d — це фіксована різниця між попереднім та наступним.

Степан, дізнавшись про проведення III етапу Всеукраїнської олімпіади з інформатики, вирішив запропонувати учасникам непросту, цікаву задачу, на тему "Арифметична прогресія". Він бере довільне додатне число A і виписує на дошці арифметичну прогресію з першим членом рівним A і різницею, рівною також A , тобто маємо послідовність $A, A+A, A+2A, A+3A, \dots$. Степана цікавить перше число в даній послідовності, яке є повним кубом деякого натурального числа. Степан довів, що для любого натурального числа A в описаній вище арифметичній прогресії існує повний куб деякого натурального числа.

Наприклад, перший член арифметичної прогресії 2, тоді маємо виписати на дошці послідовність 2, 4, 6, 8, ... Четвертий член цієї арифметичної прогресії є повним кубом числа 2 ($8 = 2^3$).

Напишіть програму, яка для заданого числа A , визначає мінімальну кількість членів арифметичної прогресії, які потрібно виписати на дошці, щоб серед них був повний куб деякого натурального числа.

Формат вхідних даних: Єдиний рядок вхідного файлу містить одне ціле число A ($1 \leq A \leq 10^9$).

Формат вихідних даних: Вихідний файл має містити одне ціле число - мінімальну кількість членів арифметичної прогресії, які потрібно виписати на дошці, щоб серед них був повний куб.

Система оцінювання:

Рішення, які вірно працюють для $A \leq 10$ набиратимуть не менше 20 балів.

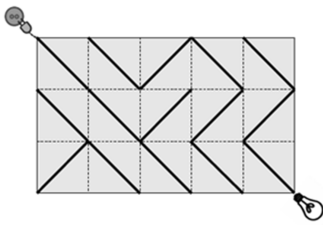
Приклади вхідних та вихідних даних:

interesting.in	interesting.out	Пояснення:
2	4	четвертий член цієї арифметичної прогресії (2, 4, 6, 8, ...) є повним кубом числа 2 ($8 = 2^3$).
4	2	другий член цієї арифметичної прогресії (4, 8, ...) є повним кубом числа 2 ($8 = 2^3$).
125	1	перший член цієї арифметичної прогресії (125, ...) є повним кубом числа 5 ($125 = 5^3$).

Задача D. Увімкніть лампу

Input file name:	lamp.in
Output file name:	lamp.out
Time limit:	1000 ms
Memory limit:	128 M

Степан розробляє електронну схему на прямокутній сітці розміром $N \times M$. Усього $N \times M$ квадратних плиток. Два (з чотирьох) протилежних кута кожної плитки з'єднані дротом.



Джерело живлення під'єднано до лівого верхнього кута сітки, лампа - до правого нижнього. Для того щоб увімкнути лампу можна повернути будь-яку плитку на 90 градусів у обох напрямках.

На зображенні лампа вимкнута. Якщо повернути будь-яку плитку у другій справа колонці, то лампа увімкнеться.

Напишіть програму, яка знаходить мінімальну кількість плиток, що треба перевернути для того, щоб увімкнути лампу.

Вхідні дані: Перший рядок містить два цілих числа N та M - розміри сітки. Далі слідує N рядків по M символів - \ або /, що характеризують напрямлення дроту на даній плитці.

Вихідні дані: Єдиний рядок вихідного файлу має містити відповідь на задачу, або ж повідомлення "NO SOLUTION", в тому випадку, коли увімкнути лампу неможливо.

Обмеження: $1 \leq N, M \leq 500$.

Система оцінювання: В даній задачі дванадцять блоків. Тести нараховуються окремо за кожен блок, при умові проходження усіх тестів блоку.

40 балів можна отримати, якщо Ваша програма вірно працює при $1 \leq N \leq 4$ та $1 \leq M \leq 5$.

Приклади вхідних і вихідних даних:

lamp.in	lamp.out
3 5 \\//\ \\/// /\\//	1

Задача Е. Свято в Ужляндії

Input file name:	holiday.in
Output file name:	holiday.out
Time limit:	1500 ms
Memory limit:	128 M

В Ужмісті - столиці Ужляндії, планується провести фестиваль Ужляндських виробів. Президент Ужляндії розуміє, що гості з усіх куточків Ужляндії прийдуть на цей фестиваль і хоче зробити їх подорож як можна менш витратною.

Ужляндія складається з N міст. Дорожня система Ужляндії не змінювалася з давніх пір, і тому вона містить лише $N-1$ двонапрямлених доріг. По цих дорогах все ще можна проїхати з будь-якого міста Ужляндії в будь-яке інше (можливо, через проміжні міста). Історично, за проїзд по кожній дорозі стягується податок. Він стягується одним з міст, що з'єднуються дорогою. Кажуть, що це місто відповідальне за цю дорогу. Величина податку може бути різною для різних доріг.

Для проїзду до столиці деяким жителям Ужляндії необхідно заплатити більше, а іншим менше. Це впливає на атмосферу свята: той, кому довелося заплатити занадто багато, приїжджає без настрою. Президент хоче мінімізувати суму дорожнього податку, яку потрібно заплатити на шляху до столиці з будь-якого міста Ужляндії. Для цього він вирішив наказати усім містам скасувати податок на одній з доріг, за які вони відповідальні. Якщо місто не відповідальне за жодну дорогу, то і скасовувати податок йому не потрібно.

Допоможіть визначити, на яких дорогах потрібно скасувати податок, щоб вартість подорожі до столиці стала якомога менша. Вартість подорожі до столиці визначається як максимальне значення для всіх міст Ужляндії.

Вхідні дані: Перший рядок вхідного файлу містить ціле число N ($2 \leq N \leq 10^5$) - кількість міст в Ужляндії. Міста пронумеровані натуральними числами від 1 до N , Ужмісто має номер 1. Далі йдуть $N-1$ рядок. Кожен з цих рядків містить три розділених одиночними пробілами цілих числа X, Y, Z : між містами з номерами X і Y є дорога, за яку відповідальне місто з номером X , і за проїзд по якій стягується податок рівний Z ($1 \leq Z \leq 10^4$).

Вихідні дані: Єдиний рядок вихідного файлу має містити одне ціле число - мінімально можливе значення суми податків на шляху з міста в столицю.

Система оцінювання:

$N \leq 10$ – не менше 20 балів

$N \leq 20$ – не менше 30 балів

$N \leq 200$ – не менше 50 балів

$N \leq 2000$ – не менше 70 балів

Приклади вхідних і вихідних даних:

holiday.in	holiday.out
3	2
1 2 2	
1 3 3	
5	1
1 2 1	
1 3 1	
3 4 1	
3 5 1	