Задача A. Foo Bar 2

Ели и Cel1o не обичат обикновени игри, понеже са много под интелектуалното им равнище. Дори играта **foo bar** се оказа под нивото им. Затова, играят следната усложнена версия, подпомагаща хидратирането им – намислят цяло положително число X и започват да изреждат естествените числа подред, започвайки с X. Всеки път, когато някой от тях трябва да каже число A, делящо се на 3, той заменя това число с думата `foo`, повторена толкова пъти, колкото е **максималната степен** на 3, деляща A. Например, вместо числото 54 те казват `foo foo foo`. По подобен начин, всеки път когато трябва да кажат число B, делящо се на 5, те заменят B с толкова повторения на думата `bar`, колкото е **максималната степен** на 5, деляща B. Например, вместо 100 казват `bar bar`. Когато дадено число се дели и на 3 и на 5, те казват и `foo`-тата и `bar`-овете, като първо казват всички `foo`-та и после изреждат всички `bar`-ове. Така, вместо 120 те казват `foo bar`. Всеки път когато някой сбърка – изпива чаша минерална вода на екс.

Мими обича да регистрира, случващото се, като си записва изречените `foo`-та и `bar`-ове, от началото на играта до нейния край. Нито Ели, нито Cel1o си спомнят какво точно е ставало при предната игра и твърдят, че не са сбъркали нито веднъж. Мими е категорична, че не е така, и това личи от записките ѝ. Необходима е програма, която да проверява каква е истината. Както се досещате, и тримата страдат от главоболие и затова тази отговорна задача се пада на вас.

Вход: На първия ред на стандартния вход е зададен броят T на тестовете. Всеки тест се състои от 2 реда. На първия ред е зададено цялото N – броят думи, които Мими е записала за конкретната игра ($1 \leq N \leq 1000$). На следващия ред са записани N думи, разделени с по един интервал. Всяка от думите е или `foo` или `bar`.

Изход: За всеки тест, програмата ви трябва да изведе на отделен ред думата `Yes`, ако е възможно никой да не е сбъркал в цялата игра, а в противен случай – думата `No`. Обърнете внимание че Мими, Ели и Cel1o знаят наистина големи числа (нали са вишисти).

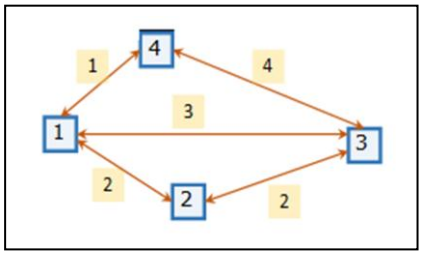
Пример:

Вход	Изход
2 14 bar foo foo bar bar foo foo foo foo bar foo bar foo foo 10 foo bar foo bar foo bar foo bar foo bar	Yes No

Пояснение: В първия тест е възможно Ели и Cel1o да са започнали с числото 20 и последното число което са казали да е 36. Тогава те са казали последователно следните неща: `bar` вместо 20, `foo` вместо 21, `foo` вместо 24, `bar bar` вместо 25, `foo foo foo` вместо 27, `foo bar` вместо 30, `foo` вместо 33, `bar` вместо 35 и накрая `foo foo` вместо 36.

ePay.bg ICB SOFTWARE INNOVATION (h[s]) HYPERSCIENCE STEP IT ACADEMY MusalaSoft SBTech we know sports д-р Владимир Алексиев

Задача В. Аварийна група



Електроразпределителното дружество „BEZ“ решава да подмени аварийните автомобили с електромобили. Обектите, за които отговаря аварийната група, в която работи Петър, са разположени в няколко съседни селища. На картата на района са означени селищата, пътищата, които ги свързват и дължините на пътищата. Между всяка двойка населени места съществува най-много един двупосочен път. Няма пътища, водещи от населено място директно към самото него. След като се получи сигнал за авария, Петър трябва да състави списък с различните маршрути до селището на аварията, дължината на които не надвишава разстоянието, което електромобилът може да измине без презареждане. Маршрутът трябва да е само за отиване до мястото на аварията и да е такъв, че да минават през дадено населено място не повече от веднъж. Напишете програма, която изготвя желания списък от маршрути, дължините на които не превишават максималния пробег на електромобила.

Вход: При едно изпълнение, програмата трябва да обработва няколко тестови примера. За всеки тестов пример, от първия ред на стандартния вход се въвеждат две цели числа, разделени с интервал – броят V на населените места в района на аварийната група, номерирани от 1 до V и броят R на пътищата между двойки населени места ($2 \le V \le 64$, $2 \le R \le 64$). От следващите R реда се въвеждат по три цели числа C_1, C_2 и D. C_1 и C_2 са номерата на две населени места, свързани с път, а D – разстоянието между тях ($1 \le D \le 9999$). От последния ред на теста се въвеждат три цели числа: номерът S на населеното място, от което тръгва аварийната група, номерът T на населеното място, в което е възникнала аварията и дължината M на пътя, който електромобилът може да измине без презареждане.

Изход: За всеки тест, на отделни редове на стандартния изход, програмата трябва да изведе намерените маршрути, подредени в нарастващ ред на дължините им, а при равни дължини – в лексикографски нарастващ ред. Редът започва с дължината на маршрута, следвана от ‘:’ и интервал. Следват номерата на населените места, които формират маршрута, разделени с интервали. Ако няма намерени маршрути, които удовлетворяват условията, програмата трябва да извде ред с думата **No**.

Пример:

Вход	Изход
4 5 1 2 2 1 3 3 1 4 1 2 3 2 3 4 4 1 3 4 4 5 1 2 2 1 3 3 1 4 1 2 3 2 3 4 4 1 4 10	3: 1 3 4: 1 2 3 1: 1 4 7: 1 3 4 8: 1 2 3 4

ePay.bg ICB SOFTWARE INNOVATION (h[s]) HYPERSCIENCE STEP IT ACADEMY MusalaSoft SBTech we know sports д-р Владимир Алексиев

Задача C. Средно аритметично

Няма нищо по-просто от това да се намери средното аритметично на зададен брой цели числа. Но ако не знаеш броя на числата, май не е съвсем лесно. Напишете програма която да намира средното аритметично на зададени числа.

Вход: На един ред на стандартния вход ще бъдат зададени числата, чието средно аритметично трябва да се намери. Единственото нещо, което знаем за числата е че са от тип `int`.

Изход: На стандартния изход, програмата трябва да изведе намереното средно аритметично с пет знака след десетичната точка.

Пример

Вход	Изход
3 100024 77912 5 101	35609.00000

ePay.bg ICB SOFTWARE INNOVATION (h[s]) HYPERSCIENCE STEP IT ACADEMY MusalaSoft SBTech we know sports д-р Владимир Алексиев

Задача D. Паралелепипед

Даден е паралелепипед $P = \{(x,y,z): x_0 \le x \le x_1, y_0 \le y \le y_1, z_0 \le z \le z_1\}$ и две точки (x_2, y_2, z_2) и (x_3, y_3, z_3). Координатите на върховете на паралепипеда, както и на двете точки са цели числа от интервала [-1000, 1000]. Напишете програма, която проверява дали дадените две точки лежат върху повърхността на паралелепипеда и ако е така, пресмята квадрата на дължината на най-късия път, минаващ по повърхността на паралелепипеда и съединяващ двете точки.

Вход: На първия ред на стандартния вход ще бъде зададен броят на тестовете. За всеки тест, на първия ред са зададени числата x_0, x_1, y_0, y_1, z_0, z_1, а на втория – числата x_2, y_2, z_2, x_3, y_3, z_3, разделени с по един интервал.

Изход: За всеки тестов пример, на отделен ред на стандартния изход програмата трябва да изведе квадрата на разстоянието между двете точки. Ако поне една от дадените точки не лежи върху повърхността на паралелепипеда, тогава на съответния ред програмата трябва да изведе –1.

Пример

Вход	Изход
2 0 3 0 4 0 5 1 1 0 2 2 0 0 1 0 1 0 1 0 0 0 2 2 2	2 -1

ePay.bg ICB SOFTWARE INNOVATION (h[s]) HYPERSCIENCE STEP IT ACADEMY MusalaSoft SBTech we know sports д-р Владимир Алексиев

Задача Е. Три триъгълника

Дадени са две цели числа P и Q. Напишете програма която изчертава със знака звезда (`'*'`) фигура, състояща се от два равнобедрени триъгълника с бедра от по P звезди и един равнобедрен триъгълник с бедра от по Q звезди, лежащ между другите два, както е показано в примерния изход. Двата триъгълника с бедра с дължина P лежат на основите си, които се намират в един ред, а звездата в десния край на основата на левия триъгълник съвпада със звездата в левия край на основата на десния триъгълник. Триъгълникът с бедра с дължина Q е „спуснат“, с върха срещу основата надолу, в междината между двата други триъгълника така, че звездата във върха му съвпада с общата звезда на другите два триъгълника, а бедрата му съвпадът с бедрата на другите два триъгълника там, където се опират един в друг.

Вход: На първия ред на стандартния вход ще бъде зададен броят на тестовете. За всеки тест, на единствен ред на стандартния вход са зададени числата P и Q, $3 \leq P, Q \leq 80$.

Изход: За всеки тестов пример, програмата трябва да изведе на стандартния изход исканата фигура. Най-левите знаци `'*'` на фигурата трябва да се намират в първа позиция на съответния ред.

Пример

Вход	Изход

1
4 6
	`  **********`
	`   *          *`
	`    *        *`
	`   *  *    *  *`
	`  *    *  *    *`
	`**********`

Задача F. Неотрицателни частични суми

Дадена е редица от N цели числа $a_1, \ldots, a_N$. Резултатът от циклично завъртане на редицата на k стъпки наляво ($0 \le k \le N - 1$) е редицата: $a_{k+1}, a_{k+2}, \ldots, a_N, a_1, a_2, \ldots, a_k$. Напишете програма, която да определи колко от редиците, получени като резултат от циклични завъртания на дадената редица наляво, удовлетворяват условието: сумите на първите i числа в редицата да са по-големи или равни на нула за всяко i в интервала $[1, N]$?

Вход: Програмата трябва да обработва няколко тестови случая, зададени на стандартния вход. Данните за всеки тестов случай са зададени на два реда. На първия от тях е зададено цялото число N – броят на числата в редицата ($1 \le N \le 10^7$), а на втория – елементите на редицата $a_1, \ldots, a_N$ ($-1000 \le a_i \le 1000$). Входните данни завършват с ред, съдържащ числото 0.

Изход: За всеки тест, програмата трябва да изведе на отделен ред на стандартния изход едно цяло число – броя на намерените редици, удовлетворяващи поставеното условие.

Пример

Вход	Изход
3 2 2 1 3 -1 1 1 1 -1 0	3 2 0

Задача G. Скрити числа

Запомнянето на акаунти, пароли, PIN кодове, тайни думи и други средства за идентификация, персонализация, защита на данни и какво ли още не, е непосилна задача за всеки нормален човек. Записването им в Notes на мобилния, или в хартиено тефтерче в явен вид, очевидно, носи големи рискове. Затова Пламчо, като виден програмист, решил да си направи собствена система за криптиране на низове. За целта той използва две „тайни числа“, които скрива по следния начин. Генерира "случаен" низ от нули и единици, в която двете скрити числа са дължините на най-дългия подниз от нули и най-дългия подниз от единици. Например, числата 4 и 5 може да се скрият в низа `111110000`, в низа `10111110100001`, както и в много други низове. Напишете програма за намиране на дължините на най-дългите поднизове от нули и единици на зададен двоичен низ.

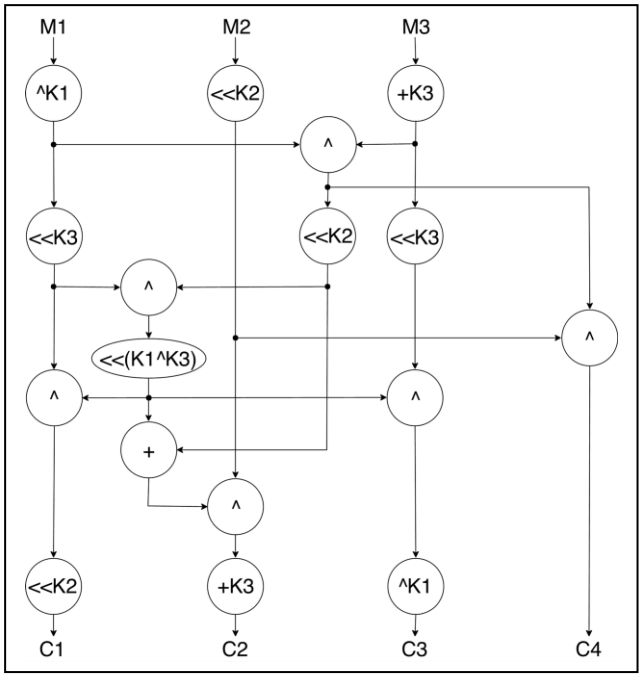
Вход: На отделни редове на стандартния вход ще бъдат зададени низове, състоящи се от единици и нули.

Изход: За всеки от зададените низове програмата трябва да изведе на отделен ред на стандартния изход дължината на най-дългия му подниз, съставен само от нули, и дължината на най-дългия му подниз, съставен само от единици.

Пример

Вход	Изход
`111000110011111` `1101`	`3 4` `1 2`

Задача Н. Хакер



Налага се да проверите сигурността на вашата държава, като опитате да разбиете схемата за кодиране на националната агенция за сигурност. Имате няколко ключа и кодирани съобщения, но задачата изглежда невъзможна, тъй като нямате алгоритъма за декриптиране. По една щастлива случайност, късно през нощта, намирате на бюрото на шефа си диаграма на кодиращата схема (показана на фигурата) и откривате, че тя е лесна за разбиване. Напишете съответна програма за дешифриране.

Всяко 72-битово съобщение M, разделено на три 24-битови групи, означени отляво на дясно с $M1$, $M2$ и $M3$, се кодира в 96-битова шифрограма C, разделена на четири 24-битови групи – означени отляво на дясно с $C1$, $C2$, $C3$ и $C4$, използвайки 72-битов ключ K, разделен на три 24-битови групи – $K1$, $K2$ и $K3$. Кодирането използва три операции над 24-битови операнди:

- ***изключващо-или***, обозначено със знака ^;
- ***събиране по модул 2^24*** (стандартно събиране, игнориращо преноса към 25-тия бит, с който и без това не разполагаме), обозначено със знака +;
- ***циклично побитово изместване на ляво***, обозначено с $<<K$, където K е броят на изместените битове (резултатът от операцията `110101010001010011101010<<7`, например, е `100010100111010101101010`).

Вход: На първия ред на стандартния вход ще бъде зададен броят T на тестовете. На всеки от следващите T реда ще са зададени по едно кодирано съобщение и ключът, с който е било кодирано – два низа с 24 и 18 шестнадесетични цифри, съответно, разделени с един интервал.

Изход: За всеки тестов пример, програмата трябва да изведе на отделен ред на стандартния изход низ, съдържащ 9-те ASCII символа на дешифрираното съобщение. Дешифрираните ASCII кодове са винаги съставени от печатуеми знаци (ASCII кодове от 0x21 до 0x7e). Всеки ASCII знак съответства на 8-те старши, средни или младши бита на всяка от групите M1, M2 и M3 в този ред, като най-левият ASCII символ съответства на старшите 8 бита, вторият – на средните 8 бита, а третият – на младшите 8 бита на групата.

Пример

Вход	Изход
2 971871216a9347c0eec722f9 953738ecc01feea4de f3e3efdf6b318e15d36f7876 c37c26932a670b14d3	BEAUTIFUL IMPORTANT

ePay.bg ICB SOFTWARE INNOVATION (h[s]) HYPERSCIENCE STEP IT ACADEMY MusalaSoft SBTech we know sports д-р Владимир Алексиев

Задача I. Стабилен полет

При полет на самолет от особена важност за сигурността на пасажерите и екипажа е поддържането на сравнително постоянна височина. Затова, прецизна апаратура непрекъснато следи височината на полета в метри и записва всяка секунда получените резултати. В зависимост от метеорологичните условия за всеки полет се определя константа H и един интервал от времетраенето на полета се нарича *H-стабилен*, ако абсолютната стойност на разликата на кои да е две измервания в интервала не надвишава H. Зададени са N-те измервания на височината на един полет и константата H. Напишете програма, която да определи дължината на най-дългия H-стабилен интервал на този полет.

Вход: На първия ред на стандартния вход ще бъде зададен броят на тестовете. Първият ред на всеки тест ще съдържа целите числа N и H ($5 \leq N \leq 1000000$, $3 \leq T \leq 1000$), разделени с един интервал, а вторият ред – N-те измервания по време на полета. Измерванията са цели положителни числа, не надвишаващи 10000, също разделени с по един интервал.

Изход: За всеки тест програмата трябва да изведе на отделен ред на стандартния изход дължината на най-дългия H-стабилен интервал по време на полета.

Пример

Вход	Изход
1 9 3 1005 1001 1003 1005 1008 1006 1006 1009 1010	4

Задача J. Игра

Иван и Петър обичат нестандартни игри. Неотдавна те попаднали в изоставена мина състояща се от N *зали,* номерирани от 1 до N, и $N - 1$ *галерии*. Всяка галерия свързва някои две от залите, никои две зали не са свързани с повече от една галерия и от всяка зала може да се стигне до всяка друга зала по единствен начин. Играта се състои в следното. Иван се скрива в зала с номер X, а Петър трябва да открие къде се намира Иван, като избере едно множество S от зали, а Иван му съобщи каква е дължината на пътя от X до всяка от залите в S, изразена в брой галерии, които образуват съответния път. Разбира се, че Петър трябва да избере S така, че да е с колкото може по-малко елементи. Напишете програма, която по зададен план на мината да определи броят на елементите в едно минимално множество, с което Петър да може да открие Иван.

Вход: На първия ред на стандартния вход ще бъде зададен броят T на тестовите случаи. Всеки тест започва с ред с броя N на залите ($5 \leq N \leq 100$). На всеки от следващите $N - 1$ реда ще бътат зададени, разделени с един интервал, двата края на една от галериите.

Изход: За всеки тестов случай, програмата трябва да изведе на отделен ред на стандартния изход намерения минимален брой зали, с помощта на които Петър да открие Иван.

Пример

Вход	Изход
1 6 1 6 2 5 3 4 4 6 5 6	2

Задача K. Палиндром

Иван и Петър основават нова, високотехнологична компания. Разбира се, една от основните им грижи е изборът на добро име за компанията. *Палиндромите*, низове които се четат еднакво в двете посоки, са известни с красотата си и много им се иска името на компанията им да е палиндром. За съжаление, не успяват да измислят привлекателно име на компанията, което да е палиндром. Затова биха желали поне телефонният номер на компанията им да бъде палиндром. При това положение компромисното решение е да изберат името на фирмата така, че когато това име се избере с буквите, отпечатани на цифровите клавиши на телефона, резултатът да е телефонен номер, който е палиндром. На стандартна телефонна клавиатура, цифровите клавиши съдържат букви както следва : на клавиш 2: ABC, на клавиш 3: DEF, на клавиш 4: GHI, на клавиш 5: JKL, на клавиш 6: MNO, на клавиш 7: PQRS, на клавиш 8: TUV и на клавиш 9: WXYZ. Напишете програма, която по зададено име на компания да проверява дали полученият от него телефонен номер е палиндром.

Вход: Програмата трябва да може да обработва няколко имена. На първия ред на стандартния вход ще бъде зададен броят T на тестовите случаи. Всеки от следващите T реда ще съдържа име, което е низ с дължина най-много 20 знака. Знаците могат да бъдат главни и/или малки латински букви. Всяка малка буква се набира със същия клавиш като съответната голяма.

Изход: За всеки тестов случай, програмата трябва да изведе на отделен ред на стандартния изход думата YES, ако полученият от името телефонен номер е палиндром или NO, в противен случай.

Пример

Вход	Изход
2 ANBOBNA iAmACoolCompany	YES NO

Задача L. Общи делители

Дадена е редица от N различни цели положителни числа. Напишете програма, която да сортира всички ненаредени множества от две различни числа на редицата във възходящ ред на броя на техните общи делители. Ако броят на общите делители на две двойки е равен, по-напред в сортирането да е двойката, в която по-малкото число е по-малко, а ако и то е еднакво в двете двойки, по-напред в сортирането да е двойката, в която по-голямото число е по-малко.

Вход: На първия ред на стандартния вход ще бъде зададен броят на тестовите примери. Всеки тестов пример започва с ред, съдържащ броя N на числата в редицата ($1 \leq N \leq 300$). На i-тия от следващите N реда ще бъде зададено поредното число K_i на редицата ($1 \leq K_i \leq$ 100000000000000 = 1.0E+14).

Изход: За всеки тест програмата програмата трябва да изведе на отделни редове на стандартния изход всички различни ненаредени двойки числа на редицата, сортирани по указания по-горе начин. В рамките на един ред първо да се изведе по-малкото число от двойката, после по-голямото.

Пример

Вход	Изход
1 4 64 18 16 12	16 18 18 64 12 16 12 64 12 18 16 64

ePay.bg ICB SOFTWARE INNOVATION (h[s]) HYPERSCIENCE STEP IT ACADEMY MusalaSoft SBTech we know sports д-р Владимир Алексиев