

Задача А: Компресия

На нашия инженер Тунчо все не му стигало мястото на харддиска и решил да си освободи малко място за научнопопулярни филмчета ;-) като компресира файловете си. Компресията, която решил да използва преобразувала едно естествено a в друго a' , състоящо се само от една цифра. Компресията изпълнява поредица от стъпки, всяка от които представлява замяната на числото със сумата от неговите цифри. След краен брой операции се достига до едноцифреното число a' .

Например, ако е дадено числото 86, след една стъпка на компресията ще се получи числото $8 + 6 = 14$. След още една стъпка на компресията ще получим крайния резултат $1 + 4 = 5$.

Помогнете на нашия мързелив инженер да си освободи място като напишете програма, която да компресира числата му.

Вход

На първия ред от стандартния вход е записано число показващо броя тестови примери N .

На следващите реда, е записано по едно положително число a което трябва да компресирате.

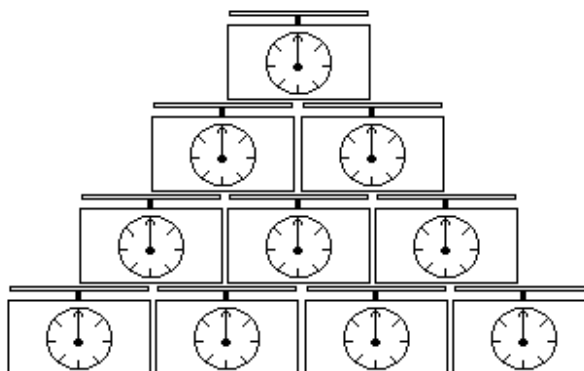
Изход

На стандартния изход трябва да отпечатате по едно число за всеки от тестовите примери – резултатът от компресията му.

Примерен ВХОД	Примерен ИЗХОД
3	7
43	3
111	1
57871	

Задача В: Скука

Нашия герой инженер Тунчо му беше толкова скучно, че реши да направи малко пролетно почистване в кабинета си. В склада изрови някакви стари теглилки, от които си построи скулптура. Тя беше съставена само от еднакви теглилки, отчитащи измереното тегло на кръгъл циферблат. Скулптурата изглеждаше като пирамида (вж. Фигурата). Тегликата на върха беше поставена така, че тежестта ѝ да се разпредели поравно между двете стоящи отдолу теглилки. Те от своя страна лежаха върху три теглилки, така че за всяка от тях тежестта се разпределяше поравно между двете стоящи отдолу. Трите теглилки лежаха върху нови четири, така, че тежестта на всяка от тях се разпределяше по същия начин и т.н. Незнайно защо, художникът беше закрепил неподвижно стрелките на всички теглилки на нулевото деление, както е показано и на фигурата. Напишете програма, която да определя какво би показвала стрелката на някоя от теглилките, ако не беше закрепена.



Вход

На първия ред от стандартния вход е записано число показващо броя тестови примери.

Всеки следващ ред описва данните по един тестов пример и се състои от три цели положителни числа T , N и M , разделени с по един интервал. Числото T е собственото тегло на всяка една от теглилките ($1 \leq T \leq 60000$), N – броят на редовете на пирамидата ($3 \leq N \leq 30$), а M – поредният номер на теглилка в най-долния ред на пирамидата, броени отляво надясно.

Изход

За всеки тестов пример програмата трябва да изведе на стандартния изход показваното от M -тата теглилка на последния ред тегло във формата на нормализирана обикновена десетична дроб (числителят и знаменателят разделени със знака за делене /).

Примерен ВХОД	Примерен ИЗХОД
1 10 4 2	85/4

Задача С: Маршрутки

В града има N спирки на маршрутни таксите ($2 \leq N \leq 1000$), номерирани с числата от 1 до N . Инженера Тунчо се намира на спирка 1 и иска да отиде до спирка N , като плати минимална цена. За целта може да използва една или повече маршрутки. В града има M маршрутни линии ($1 \leq M \leq 20\,000$). Всяка от тях предлага **еднопосочен превоз между две спирки**. Цената се определя от изминатото разстояние – C стотинки на километър за всички маршрутки в града ($1 \leq C \leq 100$). За i -тата маршрутка са известни числата S_i , E_i и D_i : начална спирка, крайна спирка и разстоянието в километри, съответно ($1 \leq S_i \neq E_i \leq N$, $1 \leq D_i \leq 1000$, D_i е цяло).

Тъй като задачата изглежда прекалено лесна, а и от Маршрутния Транспорт решили, че така прекалено много хора щели да научат да смятат алгоритъма на Дейкстра наум, на пазара били пуснати Далаверски Карти (ДК). Всяка ДК обезпечава определен брой безплатни километри при всяко ползване на маршрутка (да означим този брой с k). Цялото k е отпечатано на ДК и се избира от клиента при закупуването на ДК. ДК също струват пари: цената на ДК с k безплатни километра е $P * k$ стотинки ($C \leq P \leq 10\,000$, P е цяло). Схемата е следната: след като се качи в някоя маршрутка, пътникът не плаща първите k километра от пътя (ако целия път не надвишава k километра, пътникът не плаща нищо). ДК важат за всички маршрутки и могат да се ползват произволен брой пъти (закупуването е само веднъж). Тунчо се чуди как точно да пътува и дали да си купи ДК (и ако да, каква). Помогнете му: сметнете колко стотинки най-малко ще трябва да плати. Избраният път не е от значение – Тунчо не бърза.

Вход

Програмата трябва да обработи няколко тестови примера. Първият ред на стандартния вход съдържа броя T на тестовете. Първият ред на всеки тест съдържа числата N , M , C и P , разделени с интервали. Всеки от следващите M реда описва един маршрут и съдържа числата S_i , E_i и D_i , разделени с интервали. Всички маршрути са еднопосочни и може да има повече от един маршрут между две спирки. Гарантирано е, че има път от спирка 1 до спирка N .

Изход

За всеки тест програмата трябва да изведе на стандартния изход минималната цена (в стотинки), която Тунчо трябва да плати, за да стигне от спирка 1 до спирка N . Цената включва разходите за ДК (ако има такава) и парите, платени в маршрутките.

Примерен ВХОД	Примерен ИЗХОД
2	6
5 5 1 3	8
1 2 5	
1 3 2	
2 3 1	
3 4 3	
4 5 1	
7 8 1 4	
1 2 4	
1 4 2	
2 3 4	
2 5 5	
3 4 1	
4 5 2	
5 6 2	
6 7 2	

Задача D: Nimper!

Преди много време, още когато нямало PS и Xbox, Nimper била една от любимите игри на ученолюбивите студенти в МЕИ-то. Nimper се играе от двама души. Играта се започва с определен брой купчинки от стари процесори и се играе на 2 етапа. По време на първия етап играч 1 може да вземе част от купчинките или нито една, а след това и играч 2 да вземе част от купчинките или нито една. Никой от двамата няма право да взема всички купчинки.

При втория етап играчите се редуват. Всеки от тях си избира купчинка и може да вземе произволен брой процесора от нея. Играчът, който вземе последния процесор – печели.

Имате броя купчини и броя процесори в тях. Трябва да намерите минималният брой процесори, които първият играч трябва да вземе по време на първия етап за да бъде сигурен, че ще спечели играта.

Вход

Първият ред от стандартния вход съдържа броя тестове.

На всеки следващ ред има едно естествено число ($1 \leq n \leq 100$), което задава броя купчини. Следват n естествени числа, показващи броя процесори във всяка от купчините.

Изход

На стандартния изход отпечатайте минималният брой процесори, които първият играч трябва да премахне по време на първия етап за да може да спечели играта.

Примерен ВХОД	Примерен ИЗХОД
1 3 1 2 3	1

Задача Е: Преобразувател на числа

В тази задача, имате две положителни цели числа a и b , следвани от n на брой числа c_1, c_2, \dots, c_n . Използвайки четирите основни аритметични операции (събиране, изваждане, умножение и деление (с закръгление надолу за да се получи цяло число)) и само дадените числа – преобразувайте a в b .

Например ако са дадени $a = 2$ и $b = 10$, а $c_1 = 2, c_2 = 3$. Има два различни начина за преобразуването на a в b :

1. Първо да се умножи a с c_2 и след това 2 пъти да се прибави c_1 .
2. Да се прибави c_1 4 пъти към a .

Трябва да предвидите и още едно условие, а именно това, че различните аритметични операции имат различна цена. Цените са представени в таблица $n \times 4$, в която клетките $(i, 1), (i, 2), (i, 3), (i, 4)$ от тази таблица съдържат цените за съответно събиране, изваждане, умножение и деление с числото c_i като втори операнд. Вие трябва да намерите минималната цена за преобразуване на a в b , при условие, че никога не получавате число по-голямо от 10^5 или по-малко от нула.

Вход

На първия ред от стандартния вход се задава броя тестови примери. За всеки тестов пример на един ред се задават 3 естествени числа $1 \leq a, b \leq 10^5$ и $1 \leq n \leq 10$. На следващия ред следват n цели положителни числа c_1, c_2, \dots, c_n . След това на следващите n реда има по 4 положителни цели числа, описващи цените на операциите.

Изход

За всеки тестов пример трябва да отпечатате на нов ред на стандартния изход минималната цена за преобразуване или думата IMPOSSIBLE, ако такова преобразуване не е възможно.

Примерен ВХОД	Примерен ИЗХОД
1 45 768 3 6 2 12 1 2 3 4 1 2 3 4 4 3 2 1	7

Задача F: Фафли

Една компания произвежда на ден N на брой Фафли, които трябва да бъдат пакетираны в B кутии, всяка от които съдържа L Фафли. Цикъла на производство продължава D дни отговаря на следните критерии:

- параметрите на производството N , B ($B > 1$) и L остават константни за целия производствен цикъл;
- Фафлите произведени през един ден имат различни опаковки;
- има точно N вида опаковки, които се използват в целия производствен цикъл;
- в края на производствения цикъл, за всяка двойка от опаковки трябва да има само една кутия, която да съдържа 2-те Фафли с тези опаковки.

Например, ако всеки ден се произвеждат 9 Фафли, производствения цикъл трае 4 дена, тогава компанията може да използва 3 кутии, съдържащи по 3 Фафли всеки ден. Всичките 9 вида опаковки могат да бъдат използвани по следния начин:

	Вох #1	Вох #2	Вох #3
Day #1:	123	468	579
Day #2:	145	278	369
Day #3:	167	249	358
Day #4:	189	256	347

Напишете програма, която при зададено число N , определя стойността на L така че стойността на $D - B$ е минимална. Ако има повече от едно решение, трябва да изберете това, за което стойността на L е най-голяма.

Вход

Първият ред на стандартния вход съдържа броя T на тестовете.

На следващите T реда се съдържа по едно естествено число N ($2 \leq N \leq 2.000.000.000$), отговарящо на броя Фафли произвеждани всеки ден за съответния тестов пример.

Изход

На стандартния изход трябва да се изведат T на брой реда, всеки съдържащ по 3 числа, разделени с интервал, задаващи намерените от вас стойности за L , B и D . Ако няма решение за дадения пример, изходния ред за него трябва да съдържа 0 0 0.

Примерен ВХОД	Примерен ИЗХОД
1 300000	120 2500 2521