A. Info board

Информационно табло има правоъгълна форма с широчина **W** и височина **H** см. Горният ляв ъгъл на таблото е избран за начало на правоъгълна координатна система. Хоризонталната ос X е разположена по широчината на таблото и сочи надясно, а вертикалната ос Y – по височината на таблото и сочи надолу.
На таблото са поставени **N** съобщения. Всяко съобщение е написано на правоъгълен лист с широчина A_i и височна B_i см, **i** = 1,2,…, **N**, и е поставено така, че страните му са успоредни на краищата на таблото, а върховете му са разположени в точки с целочислени координати.
Напишете програма, която пресмята броя на различните начини, по които ново съобщение с широчина **U** и височина **V** см, може да се разположи на таблото, така че страните му да са успоредни на координатните оси, а върховете му да са в точки с цели координати. Новото съобщение не трябва да излиза от таблото, може да се допира до някои от старите съобщения, но не и да покрива техни части.

Вход
На първия ред на стандартния вход е зададен броят **T** на тестовите примери. Всеки пример започва с ред, с две цели числа – широчината **W** и височината **H** на таблото. На втория ред е зададен броят **N** на наличните съобщения. Следват **N** реда с по четири цели числа X_i, Y_i, A_i, B_i – координатите на горния ляв ъгъл, широчината и височината на всяко листче, съответно. На последния ред за всеки пример са зададени целите числа **U** и **V** – широчината и височината на новото съобщение.

Изход
За всеки тестов пример на отделен ред на стандартния изход програмата трябва да изведе намерения брой начини, по които може да се разположи новото съобщение на таблото или 0, ако няма място за него.

Ограничения

- $1 \le T \le 15$
- $10 \le U \le W \le 120$
- $10 \le V \le H \le 80$
- $0 \le N \le 16$
- $0 \le X_i < X_i + A_i \le W$, $i = 1, 2, \ldots, N$
- $0 \le Y_i < Y_i + B_i \le H$, $i = 1, 2, \ldots, N$

Примерен Вход	*Примерен Изход*
1 12 8 2 2 0 3 3 6 1 5 4 2 4	13

B. Escape

Ели е затворена в дълъг коридор, в двата края на който има врати към външния свят. Освен нея, в коридора има няколко препятствия (стени, бандити, тонколони, дънещи Джъстин Бийбър и др.). За щастие, тя носи със себе си портативен гранатомет, с който може лесно да премахне всяко от препятствията. Ели се пита, колко най-малко препятствия трябва да взриви, за да се измъкне от коридора. Помогнете ѝ, като напишете програма, която да пресмята този брой

Вход

На първия ред на стандартния вход ще бъде зададен броят **T** на тестовете. Всеки от следващите **T** реда ще съдържа описание на един коридор – низ с дължина L_i, съставен от ASCII знаци от множеството {'.', 'E', '#'}, където '.' означава празно пространство, 'E' е полето, в което се намира Ели, а '#' означава препятствие.

Изход

За всеки от тестовете на отделен ред изпечатайте едно цяло число - минималния брой препятствия, които Ели трябва да взриви за да се измъкне.

Ограничения

- $1 \le T \le 100$
- $1 \le L_i \le 100$, $i = 1, 2, \ldots, T$

Примерен Вход	*Примерен Изход*
3 .##.#..#..E#..#.........#.. #E# ....E..#	3 1 0

***Упътване**. В първия пример, ако Ели реши да излезе през лявата врата, ще трябва да премахне 4 препятствия. Ако реши да излезе през дясната врата, препятствията ще са само 3.*

C. Elections

При пропорционална система за избори на парламент, броят на получените мандати за една партия е пропорционален на подадените за тази партия действителни гласове. Това правило, обаче, е в сила само за онези партии, които са получили повече от четири процента от общия брой гласове, т.е. само тези партии участват в пропорционалното разпределение на мандатите.
Напишете програма за разпределяне на точно двеста и четиридесет мандата между партиите преминали бариерата. Пресмятането на пропорциите се извършва с целочислено деление. По този начин е възможно да бъдат разпределени по-малко мандати, като в такъв случай президентът допълва парламента с недостигащите депутати. Ако никоя партия не събере повече от четири процента, президентът назначава всички депутати.

Вход
На стандартния вход ще бъдат зададени няколко тестови примера. За всеки пример, на първия ред е зададен броят на партиите, а на втория – броят на гласове за всяка партия. Броят на партиите е по-голям от две и по-малък от сто, а броят на гласувалите – по-голям от броя на партиите и по-малък от четири милиона. Входът завършва с ред, на който е зададена една 0.

Изход
За всеки пример, на отделен ред на стандартния изход, програмата трябва да изведе резултата на всяка партия, която влиза в парламента (получава мандати). Данните за всяка такава партия са: пореден номер на партията, (в реда, по който са наредени броят на гласовете във входа, започвайки от едно), двуеточие, брой мандати. Данните за всеки две партии са разделени с един интервал.
Ако остават мандати за допълване от президента, след резултатите на всички партии програмата трябва да изведе `president:` и броят нераздадени мандати.

Примерен Вход	*Примерен Изход*
3 300000 200000 100000 5 123000 245000 23000 1200 56000 0	1:120 2:80 3:40 1:66 2:131 3:12 5:30 president:1

Забележка. *Описаните правила не съответстват на действащия в момента в Република България изборен кодекс.*

eCommera ИНФОРМАЦИОННО ОБСЛУЖВАНЕ ePay.bg Stangaone Proxiad COMSTREAM

За разпределяне на времето между потребителите на Интернет, възникнала следната задача: Всяка секунда, във възел на мрежата може да пристигне пакет, съдържащ времето (в секунди), което трябва да изчака в този възел, преди да бъде изпратен по пътя си към получателя. Ако два пакета трябва да бъдат изпратени в едно и също време, то се изпраща пристигналият по-рано, като другият се заявява за изпращане в следващата секунда при същото правило. Напишете програма, която симулира този процес и намира в коя секунда реално ще бъде изпратен всеки от пристигналите във възела пакетите.

Вход

Програмата трябва да обработи няколко тестови примера. За всеки тест, на отделен ред на стандартния вход е зададена последователност от цели, неотрицателни числа. Всяко от числата съответства на една секунда от скалата на времето и означава продължителността за изчакване на пристигналия в съответната секунда пакет. Броенето на секундите започва от 1. Числото 0 означава, че в съответната секунда няма пристигнал пакет.

Дължината на входа, за всеки от тестовете, е за по-малко от едно денонощие. Времето за изчакване на всеки от пакетите, също е по-малко от едно денонощие. Във всеки тест е изпратен поне един пакет (т.е. на всеки ред има поне едно число, различно от 0). Всички двойки съседни числа в реда са разделени с точно един интервал. Няма интервали в началото и края на реда.

Изход

За всеки тест на отделен ред програмата трябва да изведее последователност от цели неотрицателни числа. Всяко число съответства на поредна секунда от скалата на времето, започвайки от 1 и представлява момента на пристигане на пакета, който ще бъде изпратен в тази секунда. Числото 0 означава, че в тази секунда не се изпраща пакет. Изходът за съответния тест завършва в секундата, в която е изпратен и последният пристигнал пакет.

Примерен Вход	*Примерен Изход*
2 0 0 1 2 1 0 4 3 1 0 1 3 3 7 0 0 0	0 0 1 0 4 0 5 6 0 0 0 0 4 2 3 6 0 7 8 0 0 0 0 9

Упътване. *В първия тест, в първата секунда пристига пакет с време на чакане 2. Затова в първите две секунди няма изпратен пакет. В третета секунда се изпраща пакетът, пристигнал в първата секунда. В четвъртата секунда пристига пакет с време на чакане 1, и той се изпраща в петата секунда. Следващите два пакета с време на чакане 2 и 1 трябва да се изпратят на седмата секунда, затова вторият чака още една секунда и се изпраща на осмата секунда.*

E. RPG

Пешо Пингвинът е велик магьосник. Освен това е и запален геймър, който играе много ролеви игри (http://en.wikipedia.org/wiki/Role-playing_game). Пешо и Станчо са решили да играят играта „Подземия и дракони“, като и двамата са войни и използват мечове.

Бойната система твърди, че Пешо трябва да хвърля **N**-стенно зарче (със стени надписани с 1, 2, ..., **N**), а Станчо **M**-стенно (със стени надписани с 1, 2, ..., **M**). Преди играта, се определят **L** случайни числа в интервала от 1 до **N**, включително, и **K** случайни числа в интервала от 1 до **M**, включително. Ако по време на играта Пешо хвърли със зара някое от неговите **L** числа – печели играта. Станчо пък ще спечели, ако хвърли със зара някое от неговите **K** числа. Двамата се редуват да удрят (хвърлят), докато някой спечели, като Пешо започва пръв.

Двубоят има множество рундове. Понеже Пешо е могъщ магьосник се интересува каква е вероятността той да спечели битката във всеки от тях. Напишете програма, която изчислява тази вероятност.

Вход

На първия ред на стандартния вход е зададен броя **T** на тестове, с които ще бъде тествана програмата. Следват **T** блока от вида:

```
N L M K
A1 A2 … AL
B1 B2 … BK
```

където A_1, A_2, … A_L са печелившите за Пешо, а B_1, B_2, …, B_K – печелившите за Станчо числа. Гарантирано е, че поне един от противниците може да спечели.

Изход

За всеки тестов пример програмата трябва да се изведе на стандартния изход обикновена дроб, показваща вероятността Пешо да спечели битката. Дробта трябва да бъде правилна, тоест числителят и знаменателят ѝ да са взаимно прости (неправилна дроб от вида 2/4, трябва да се замени с ½). Числата 0 и 1 трябва да бъдат представяни като 0/1 и 1/1.

Ограничения

- $1 \le N, L, M, K \le 1000$
- $1 \le A_i \le N$
- $1 \le B_j \le M$

Примерен Вход	*Примерен Изход*
2 2 1 6 2 1 1 2 49 3 52 5 2 47 29 46 40 11 20 1	3/4 78/193

F. Olympiad

В държавата на жълтите патета има **N** населени места, номерирани с числата от **1** до **N**, включително. Населените места са два типа: K града и **N** – **K** села. Столица на държавата е с номер **C** и може да е както град, така и село. В държавата има **M** пътя, които свързват пряко някои от населените места. Всеки път е двупосочен, свързва две различни населени места S_i и E_i и се преминава за T_i минути, **i** = 1,2,…,**M**. Решено било, в столицата да се проведе Първата ПОП (Патешка Олимпиада по Програмиране). За целта от столицата до всеки градове били изпратени вестоносци (по един вестоносец на град) с информация за олимпиадата. Напишете програма, която пресмята в какъв ред и след колко време всеки от вестоносците ще достигне до своя град. Предполага се, че вестоносците бързат, не спят и не се бавят никъде.

Вход

На първия ред на стандартния вход е зададен броят **T** на тестовите примери. Всеки тестов пример започва с ред, с четири цели числа **N**, **M**, **K** и **C.** Следва ред с **K** цели числа – номерата на градовете и **M** реда, на i-тия от които са зададени трите цели числа S_i, E_i и T_i. Гарантирано е, че от столицата може да се достигне до всеки град по наличните пътища (евентуално, през други населени места).

Изход

За всеки тестов пример на стандартния изход се извеждат **K** двойки числа: за всеки град трябва да бъде изведен неговият номер и минималното време, когато вестоносецът може да достигне до него (времето се измерва в минути от момента в който вестоносците са излезли от столицата). Двойките трябва да бъдат подредени по време на пристигане на вестоносеца. Ако има два вестоносеца с равно време на пристигане, по-напред се извежда този, който има по-малък номер на града, до който трябва да достигне.

Ограничения

- $2 \le N \le 20000$; $1 \le M \le 40000$; $1 \le K, C \le N$; $1 \le T_i \le 100$.

Примерен Вход	*Примерен Изход*
2 5 4 5 1 1 2 3 4 5 1 2 1 2 3 10 3 4 100 4 5 100 5 5 3 1 2 4 5 2 1 1 2 3 10 3 4 100 4 5 100 1 5 1	1 0 2 1 3 11 4 111 5 211 5 1 2 1 4 101

G. Snakes

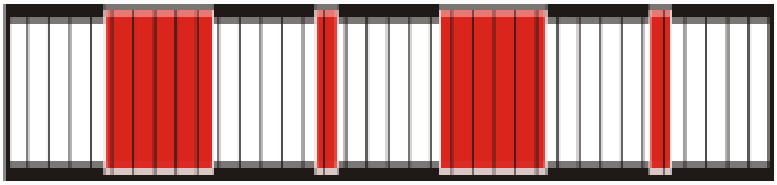
Ели не обича змии. Всъщност ги мрази. Но с повод и без повод нейните познати ѝ подаряват такива. Ели подозира, че се опитват да ѝ намекнат нещо.
Змиите са ужасни животни и биха ви нападнали без да им мигне окото (отчасти защото нямат клепачи и не могат да мигат) ако ги настъпите. Нещо повече, те биха нападнали и всяко друго нещо, което застане върху тях – включително друга змия.
Ели има дълъг, но тесен терариум (разбирай „отсечка от права“) с дължина **L** и **N** змии (също отсечки) с дължини A_1, A_2, …, A_N. Тъй като терариумът е твърде тесен, във всяка негова част може да лежи най-много една змия, без да застъпва друга (за да не се нападат). Ели има план как да оправдае изхвърлянето на известен брой от змиите! Тя планира да ги постави в терариума по такъв начин, че:

1. Никоя да не застъпва друга (възможно е обаче две змии да се допират);
2. Не може да се сложи нито една друга змия, без да се наруши правило 1.

Змиите не се движат, не могат да излизат от терариума, но могат да се докосват до стените му, а краищата им могат да бъдат разполагани дори в участъци на терариума с нецелочислени координати! Казано на по-човешки език, Ели трябва да разположи някои от **N**-те отсечки с дължини A_1, A_2, …, A_N върху отсечка с дължина **L**, така че никои две да не се застъпват, и никоя от оставащите да не може да бъде поставена изцяло в непокритите със змии частина терариума.
Помогнете на Ели. Напишете програма, която да намери максималния брой змии, чието изхвърляне може да оправдае ако разположи останалите оптимално.

Вход
На първия ред на стандартния вход ще бъде зададен броят **T** на тестовете, които програма трябва да обработи. Всеки тест се състои от два реда. На първия от тях са зададени целите числа **N** и **L** – броят на змиите и дължината на терариума, а вторият съдържа числата A_1, A_2, …, A_N – дължините на всяка от змиите.

Изход
За всеки тест, на отделен ред на стандартния изход, програмата трябва да изведе максималния брой змии, които Ели може да изхвърли.



Ограничения

- $1 \le T \le 20$
- $1 \le L \le 1000$
- $1 \le N \le 30$
- $1 \le A_i \le 100$

Примерен Вход	*Примерен Изход*
3 2 9 1 8 36 5 1 5 1 5 5 18 6 2 10 2 9 9 2	1 1 4

Упътване. *На фигурата е показано едно възможно разпределение във втория пример.*

Трудна е работата на криминалистите. След всяко престъпление трябва да възстановяват събития от миналото. На Института по криминалистика поставили следната задача: По ежедневните показания на един електромер в къща в гората, записани тайно от електрическата компания, да се определи максималния и минималния брой последователни дни в зададен интервал от време, през които къщата е стояла празна, т.е. показанията на електромера не са се променяли. По тази заявка главният математик на Института по криминалистика формулирал следния модел:

Дадена е ненамаляваща редица от цели числа A_1, A_2, ..., A_N – показанията на електромера за първия, втория, ..., **N**-тия ден и два индекса L_i и R_i. Да се намери най-дългия и най-късия интервал и от L_i-тия ден до R_i-тия ден включително, в които числата (показанията на електромера) са равни.

Вход

На стандартния вход ще бъдат зададени няколко тестови примера. Всеки пример започва с ред с две цели числа – **N** и **M**, задаващи броя показания и броя въпроси, съответно. Следващият ред съдържа **N**-те цели числа A_1, A_2, ..., A_N. Следват **M** реда със заявки, всяка от които се състои от две числа L_i и R_i, определящи интервал от време, за който да се реши задачата. Последният ред на входа съдържа единствено число 0.

Изход

За всяка заявка, програмата трябва да изведе на отделен ред на стандартния изход първо броя на срещанията на числото, което се среща *най-малък брой пъти*, последвано от броя на срещанията на числото, което се среща *най-голям брой пъти* в съответния интервал. Ако тези две числа са равни да се изведе само едното от тях, виж примера.

Ограничения

- $1 \le N \le 100{,}000$
- $1 \le M \le 400{,}000$
- $1 \le L_i \le R_i \le N$
- $-100{,}000 \le A_j \le A_{j+1} \le 100{,}000$

Примерен Вход	*Примерен Изход*
10 8 -4 -4 2 2 2 2 6 10 10 10 2 3 1 10 5 10 1 3 1 2 1 1 3 6 5 8 0	1 1 4 1 3 1 2 2 1 4 1 2

1. Periodic

Период на безкрайна периодична дроб, е (непериодична) част от дробта, която се повтаря до безкрайност. Прост пример за това е 1/3, която можем да отбележим като 0.33333... или 0.(3) ("нула цяло и три в период"). Друг пример е 1/14 = 0.071428571428571... = 0.0(714285). Забележете, че самият период е непериодичен – тоест 0.(4242) бихме могли да запишем като 0.(42), затова (4242) не е валиден период.

Подобно можем да дефинираме безкраен периодичен низ, съставен само от главни букви на английската азбука 'A' - 'Z': непериодичен низ S, повторен безкрайно много пъти, което отбелязваме с (S). Например, (ABC)=ABCABCABCA… . Например (GAGA) не е валиден периодичен низ, тъй като може да се запише като (GA) = GAGAGAGA... .

Петр е дефинирал "сума" на два безкрайни периодични низа S_1 и S_2 като безкраен низ S_3, за който $S_3[i] = S_1[i] + S_2[i]$ (тоест съдържа на всяка своя позиция сумата на съответните букви от двата безкрайно периодични низа). Буквите на латинската азбука ъбираме по следните правила. Добавяне на 'A' към буква дава в резултат следващата буква на азбуката (след 'Z' следва 'A'). Така 'A'+'A'='B', 'A'+'B'='C',…, 'A'+'Y'='Z', 'A'+'Z'='A'. Добавяне на 'B' към буква на азбуката означава да добавим 'A' два пъти: 'B'+'A'='A'+('A'+'A')='C', 'B'+'B'='D', …, 'B'+'Y'='A', 'B'+'Z'='B'. Резултатът от добавяне на останалите букви се дефинира по сходен начин (добавяне на 'C' е еквивалентно на 3 пъти добавяне на 'A', на 'D' – на 4 пъти добавяне на 'A', ..., на 'Z' - на 26 пъти, добавяне на 'A', т.е. добавянето на 'Z' дава в резултат втория операнд).

Интересен факт е, че сумата на два безкрайни периодични низа също е безкраен периодичен низ! Например (AB)+ (ZYZYZ) = (AAAAABZBZB).

Ели и Петр играят на следната игра. Единият от тях си намисля три числа **N**, **M** и **K**, а другият трябва да намери три непериодични низа S_1, S_2 и S_3, с дължина **N**, **M** и **K** съответно и такива, че $(S_1)=(S_2)+(S_3)$. Помогнете на Ели, като напишете програма, която отговаря на такива въпроси.

Вход

На първия ред на стандартния вход ще бъде зададен броят **T** на тестове. Следват **T** реда, всеки съдържащ по три цели числа **N**, **M**, и **K**.

Изход

За всеки тест, програмата трябва да изведе на стандартния изход лексикографски най-малкото решение $(S_1)=(S_2)+(S_3)$ или "NO SOLUTION" ако няма такова.

Ограничения

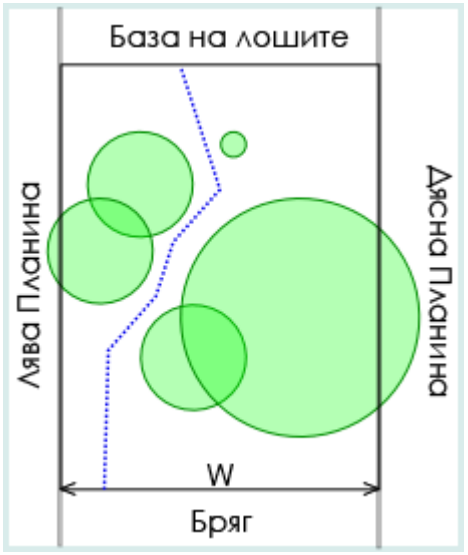
- $1 \le \mathbf{T} \le 30$
- $1 \le \mathbf{N}, \mathbf{M}, \mathbf{K} \le 20$

Примерен Вход	*Примерен Изход*
10 2 5 5 3 3	(AAAAABZBZB)=(AB)+(ZYZYZ) NO SOLUTION

J. Towers

eCommera ИНФОРМАЦИОННО ОБСЛУЖВАНЕ ePay.bg® StangaOne Proxiad COMSTREAM

Ели е отново на бойното поле! Тя е акостирала на брега на врага и иска да стигне до противниковата база, която се намира навътре в континента. За нейн ужас, както от лявата, така и от дясната ѝ страна, има високи планини, които не може да премине, а промеждутъкът между тях е осеян с **N** вражески кули. За простота можем да си представим пространството между планините като ивица с широчина **W**, в която са разположени **N** точки с целочислени координати (кулите).



Във всяка кула има определен брой войници $\mathbf{A_i}$. Колкото повече войници има в една кула, толкова по-ефективна е отбраната ѝ. Ели е изчислила, че ако в една кула има X войника, тя трябва да се придвижва на разстояние, *стриктно по-голямо* от X*X метра от кулата. Макар и обсегът на кулите да не стига до брега (тоест Ели може да се движи свободно по цялата му широчина), е възможно те да са така разположени, че момичето да не може да премине ивицата без да навлезе в обсега на някоя от тях.

За щастие, Ели е донесла със себе си своя снайпер, след всеки изстрел на който един войник по-малко ще ѝ прави проблеми. Така тя може да обезвреди произволен брой войници от всяка от кулите (включително николко или всички). Например, ако дадена кула е имала 5 войника, тоест е била с обсег 5*5 = 25 метра, тя може да отстрани двама от тях, като така намалява обсега на кулата до едва 3*3 = 9 метра.

Стрелянето със снайпера ѝ отнема време, а изпълняването на мисията ѝ възможно най-бързо е от критична важност. Сега тя се чуди какъв е най-малкият брой войници, които трябва да обезвреди, за да може да мине безпрепятствено между кулите.

Вход

На първия ред на стандартния вход е зададен броят на тестовете **T**. Всеки тест започва с ред, съдържащ целите числа **N** и **W**. Следват **N** реда, всеки от които съдържа по три цели числа $\mathbf{X_i}$,$\mathbf{Y_i}$ и $\mathbf{A_i}$ –координатите на кулата и броят на войници в нея (координатната ос координатната ос Y е контурът на лявата планина, т.е. $\mathbf{X_i}$ е разстоянието от кулата до лявата планина, а линията на брега се намира достатъчно далече на юг - примерно с големи отрицателни Y координати), така че обзорът на никоя кула не стига до там.

Изход

За всеки пример програмата трябва да изведе на стандартния изход един ред, съдържащ минималния брой войници, които трябва да отстрани Ели, за да може да премине през ивицата. Забележете, че това винаги е възможно, ако например тя премахне всички войници.

Ограничения

- $1 \le \mathbf{T} \le 30$
- $1 \le \mathbf{N} \le 30$
- $1 \le \mathbf{A_i} \le 200$
- $3 \le \mathbf{W} \le 1000$
- $1 \le \mathbf{X_i} \le W$
- $1 \le \mathbf{Y_i} \le 1000$

Примерен Вход	*Примерен Изход*
6 3 6 1 8 2 3 8 2 5 8 2 9 20 1 5 5 1 10 4 5 4 3 5 11 5 15 3 4 15 9 3 19 1 2 19 8 2 19 17 2 1 8 4 4 10 3 3 1 1 1 1 2 1 2 2 1 2 51 46 2 8 8 1 8 5 24 3 18 2 6 23 2 10 10 2 18 13 3 13 26 1	3 10 9 1 8 0

Указание. *Едно възможно решение в първия пример е да се намали броя войници в първата кула до нула, а във втората – до един. Фигурата в условието изобразява разположението и обсега на кулите от последния пример.*

eCommera ИНФОРМАЦИОННО ОБСЛУЖВАНЕ ePay.bg Stanga.one Proxiad COMSTREAM

Знаете ли, че английската дума за жаргонен език ("slang") всъщност е съкращение на израза „съкратен език“ ("shortened language")? Ели реши да съкрати няколко думи от своя речник, ползвайки следната стратегия: за да съкрати една дума тя избира *непразна* нейна поддума, която все пак не е цялата дума (тоест една или повече *последователни* букви, но не всички) и ги премахва от думата. Сега тя се пита колко възможни резултатни думи могат да се получат за всяка от нейните думи. Помогнете ѝ като напишете съответна програма.

Вход

На първия ред на стандартния вход ще бъде зададено едно цяло число **N** - броя думи, които Ели иска да съкрати. Следват **N** реда, всеки съдържащ по една дума $\mathbf{W_i}$ - последователност от малки букви от английската азбука ('a'-'z').

Изход

За всяка дума, на отделен ред на стандартния изход изведете едно цяло число - броя на различните резултати, които могат да се получат.

Ограничения

- $1 \le \mathbf{N} \le 20$
- $2 \le |\mathbf{W_i}| \le 100$

Примерен Вход	*Примерен Изход*
4 dabada gogokefakefa withoutitimjustespr spfeusefniervwethradawawspfeeeee	16 71 178 477

Указание. *От "dabada" могат да се получат думите {"abada", "bada", "ada", "da", "a", "dbada", "dada", "dda", "d", "daada", "daa", "dabda", "daba", "dab", "dabaa", "dabad"}.*