

XX РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
СОФИЯ, 9–10 МАЙ 2008

Задача за закръване. ЗАЙЦИ И КОСТЕНУРКИ

Навярно сте запознати с много феномени на костенурки, надбягващи зайци, зайци, решаващи диференциални уравнения и Станчо, присъстващ в условия на задачи по програмиране. Рядко обаче тези феномени се срещат едновременно, както в тази задача.

Един ден Станчо решил (за пореден път), че трябва да направи нещо значимо за света. Той не веднъж е писал статии по различни алгоритмични и математически проблеми. Не веднъж е пресушавал по две каси минерална вода на ден ... Напоследък, обаче, беше обзет от лошо чувство. Да – цената на минералната вода се повиши. Освен това, след гледане на „Костенурките Нинджа“ (епизода със заека самурай), Станчо бе обзет от мисли за борбата между зайците и костенурките. Сред сериозни анализи, той стигнал до извода, че въпреки, че зайците могат да скачат, костенурките имат черупки. За да продължи по нататъшния анализ, на Станчо му трябва програма която да пресмята статистики за определени групи зайци и костенурки. Според Станчо, в една група зайци и костенурки доминират тези които, са повече (зайци или костенурки). Ако броят на зайците и костенурките е равен, ситуацията е конфликтна и трябва костенурките-нинджа да се намесят, за да се нормализират нещата. Вие трябва да напишете програма която прави този анализ.

Вход

На първият ред на стандартния вход е зададено число T ($1 \leq T \leq 10$) – броят на групите за анализ. На всеки от следващите T реда е зададено описанието на една група – низ не по-дълъг от 80 знака, съдържащ само **r** и **t**, където **r** означава заек, а **t** – костенурка.

Изход

За всяка зададена група, програмата трябва да изведе на отделен ред на стандартния изход резултата от анализа на групата. Той е **Turtles!** ако костенурките доминират, **Rabbits!** – ако доминират зайците, и **Ninja!** – ако е необходима намесата на костенурките-нинджа.

Пример

Вход	Изход
3 rtttr rrrt rtrtrrtt	Turtles! Rabbits! Ninja!

Задачата е предложена от Милослав Средков

XX РЕПУБЛИКАНСКА СТУДЕНТСКАОЛИМПИАДА ПО ПРОГРАМИРАНЕ
СОФИЯ, 9–10 МАЙ 2008

Задача А. ЛИЦЕ НА ПАРАБОЛИЧЕН СЕГМЕНТ

Параболичен сегмент се нарича фигурата, ограничена от парабола и хорда, минаваща през две нейни точки. Архимед е първият учен открил метод за намиране лице на параболичен сегмент. Приблизително 2000 години по-късно е открит общ и лесен начин за намиране лица на равнинни фигури, основан на забележителната теорема на Лайбниц–Нютон.

Дадени са три точки в равнината, нележащи на една права: $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, $P_3(x_3, y_3)$, като $x_1 < x_2 < x_3$. Да се намери лицето на фигурата, ограничена от параболата, минаваща през трите точки и отсечката P_1P_3 .

Вход

Програмата трябва да прочете от стандартния вход няколко тестови примера. Всеки тестов пример съдържа на един ред стойностите на x_1 , y_1 , x_2 , y_2 , x_3 , y_3 – цели числа от интервала $[0, 100]$.

Изход

За всеки тестов пример програмата трябва да изведе на отделен ред на стандартния изход търсеното лице, закръглено до най-близкото цяло число (в тестовите примери няма такъв, резултатът от който да е еднакво близък до две различни цели числа).

Пример

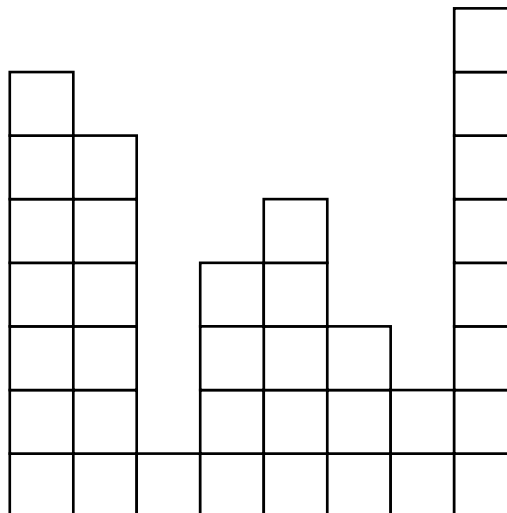
Вход	Изход
0 4 1 3 2 0	1
1 2 3 4 5 0	8
12 50 21 0 50 100	2167
10 20 50 100 88 0	4696

Задачата е предложена от Стоян Капралов

XX РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
СОФИЯ, 9–10 МАЙ 2008

Задача В. РЕДИЦА ОТ КУЛИ

Интересът към построяване на високи сгради (кули) по морския бряг нараства с всеки изминал ден. Общинската администрация на един морски курорт току що е успяла да регулира N еднакви парцела, разположени в редица край морския бряг и да ги продаде на N строителни фирми. Всяка от фирмите бързо представила проект за построяване на съответната кула, като в проекта на всяка фирма бил заложен максималният брой етажи, които фирмата може да си позволи да построи. Оказало се, че в получената редица се намират всички цели числа от 1 до N . Архитектът на курорта си имал своя идея за това как трябва да изглеждат кулите и решил да избере един интервал от редицата и да разреши строителство само на тези фирми, парцелите на които попадат в избрания интервал, като на всички фирми разреши да построят по толкова етажа, колкото е минималният брой, който някоя от фирмите в интервала е заявила. Разбира се, добре е така да се подбере интервалът, че общият брой на построените етажи да е максималният възможен. Напишете програма, която да намери този максимален общ брой етажи.



Вход

На първия ред на стандартния вход ще бъде зададен броят T на тестовите примери. Първия ред за всеки тестов пример съдържа само цялото число N ($3 \leq N \leq 40000$). На втория ред ще бъдат зададени N числа – броят на етажите които всяка от фирмите е заявила, подредени така както са подредени парцелите им на морския бряг.

Изход

На единствения ред на стандартния изход програмата трябва да изведе намерения максимален брой етажи.

Пример

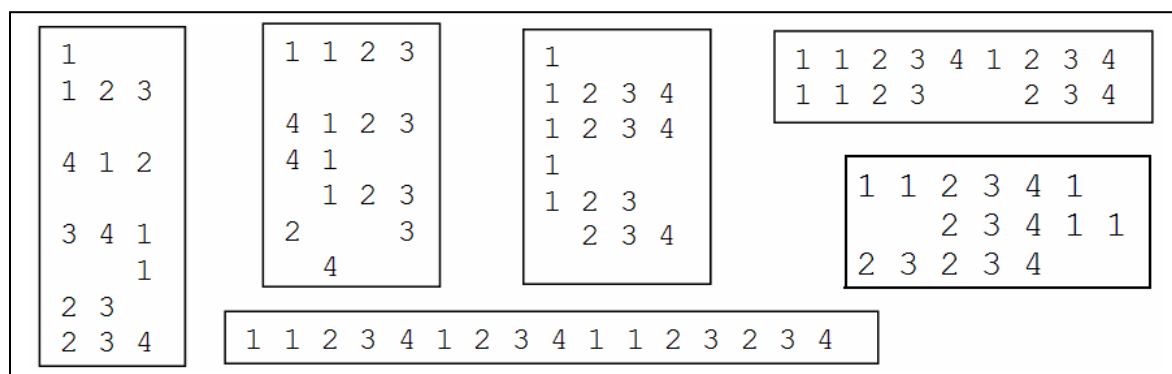
Вход	Изход
1 8 7 6 1 4 5 3 2 8	12

Задачата е предложена от Красимир Манев

XX РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ СОФИЯ, 9–10 МАЙ 2008

Задача С. СПЕЦИАЛНИ ЧИСЛА

Вероятно сте забелязали как някои хора решават, че дадено число им носи щастие, че е красиво или просто съдържа частния крипто-ключ на сайта на Външно министерство. Макар, че Станчо е над тези неща (дори и над Мерсеновите прости числа), той също смята някои числа за специални. Те разбира се са много по-специални от числата в горните примери (макар и да са цели положителни), затова той иска да ги постави в златни рамки и да ги закачи на стената си. За по-красиво, той иска да напише цифрите на всяко число в клетките на правоъгълна таблица така, че при четене на таблицата по обичайния начин да се получава съответното число. Освен това (според него) е задължително цифрите, в разположени една под друга клетки, да са еднакви. За да се получи този приятен числово-естетичен ефект, може някои клетки на таблиците да се оставят празни. Например, от 1123412341123234 може да се направят картините:



За Станчо всички тези картини са еднакво красиви. Както споменахме по-горе, обаче, рамките им са от злато. А златото все още е скъпо (дори по-скъпо от минералната вода). Зтова Станчо би предпочел периметърът на рамката да е минимален. От горните примери минимален периметър има картината най-долу вдясно. Нейният периметър е ... 4 реда и 5 колони ... въъ... Някой да си спомня формулата за периметър? Е, няма значение. И без това, за да знае Станчо колко злато да поръча, трябва да се напише програма, която за всяко от специалните му числа, отпечатва минималния възможен периметър на рамката.

Вход

На стандартния вход са зададени специалните числа на Станчо – по едно число на ред. Никое от тях не надвишава 10^{80} .

Изход

За всяко от числата на отделен ред на стандартния изход програмата трябва да отпечата минималния периметър.

Пример

Вход	Изход
1123412341123234	18
1011101011101101111111012	22
1010123	14

XX РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
СОФИЯ, 9–10 МАЙ 2008

Задача D. Максим

След като станал студент, Максим се преместил да живее в столицата, където се намира университетът, в който той учи. Сега той трябва да се погрижи за много нововъведения в живота си, като едно от важните е придвижването до различни части на града. А Максим обича всичко да прави по максимален начин, ето защо иска да открие максимално удобни маршрути до важни обекти, които му се налага да посещава. Той установил, че градският транспорт включва N ($1 \leq N \leq 10$) вида транспортни средства – автобуси, микробуси, трамваи и други. За всеки от видовете транспорт е известен броят на линиите, които той обслужва. Известен е и броят на спирките по всяка линия на градския транспорт, като общият брой на спирките е M ($100 \leq M \leq 1000$). След като се запознал с различни документи за организацията на градския транспорт в града, Максим успял да открие и максималното време в секунди, което е необходимо за придвижване на едно транспортно средство по една линия на градския транспорт от началната до крайната ѝ спирка. За различните линии то не е по-малко от 300 и не е по-голямо от 30000. Също така той разгледал карта на линиите от градския транспорт, от която намерил информация за броя K на спирките по всяка линия ($5 \leq K \leq 100$), както и номерата на самите спирки, през които преминава транспортното средство от дадена линия. Съществуват спирки, през които преминават маршрутите на различни транспортни средства. Максим проявил специален интерес към тях, тъй като от тях може да се придвижи до максимален брой различни места. Ето защо той решил, че му е необходима програма, която да познава номерата на максимално натоварените спирки, т.е. такива, през които преминават максимален брой превозни средства от градския транспорт, и да му помага да избира от тях тези, които са най-близки до дома му. Той се сетил, че сред документите за организацията на градския транспорт, видял изследване, в което се казвало, че най-натоварените спирки са с номера N и M . За щастие, тези спирки се намирали близо до дома му. Оставало само да измери времето в минути, което му е необходимо да стигне до всяка от тях.

Помогнете на Максим да определи от коя спирка може да достигне до най-голям брой обекти, които му се налага да посещава. За целта напишете програма, която по зададени N , M и времето в минути за достигане на всяка от тези две спирки, определя номера на най-подходящата за Максим спирка, като удовлетворява всички горепосочени условия.

Вход

Програмата трябва да обработва няколко тестови примера при едно извикване. На първия ред на стандартния вход е зададен броят на тестовите примери. За всеки от тях програмата чете от един ред четири цели числа – номерата на спирките M и N и времето в секунди за достигане до всяка от тях.

Изход

За всеки тестов пример програмата извежда на един ред на стандартния изход едно цяло число – номера на най-удобната, съгласно горните условия, спирка.

Пример

Вход	Изход
1 149 7 40 25	7

Задачата е предложена от Каталина Григорова

XX РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
СОФИЯ, 9–10 МАЙ 2008

Задача Е. К-ТОРКИ

Дадено е множество, състоящо се от n различни елемента ($0 < n < 20$). Напишете програма, която намира броя на всичките различни ненаредени k -торки ($0 < k < 51$) от негови елементи такива, че i -тият елемент на даденото множество участва във всяка k -торка точно $b_{i,1}$ пъти или точно $b_{i,2}$ пъти, ..., или точно b_{i,j_i} пъти.

Вход

Програмата чете данните от стандартния вход, като на първия ред е даден броят на тестовете. За всеки тест стойностите на n и k са написани на отделен ред, след който следват n реда. На първо място в i -тия от тези редове е записан броят на следващите числа в реда, които са стойностите на $b_{i,1}, b_{i,2}, \dots, b_{i,j_i}$ (цели числа, по-големи или равни на 0 и по-малки от 10).

Изход

За всеки тестов пример програмата трябва да изведе на отделен ред на стандартния изход търсения брой.

Пример

Вход	Изход
2	3
3 4	1
2 0 1	
2 0 3	
4 1 2 3 4	
3 4	
2 0 3	
2 1 2	
2 0 1	

Задачата е предложена от Емил Келеведжиев

XX РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
СОФИЯ, 9–10 МАЙ 2008

Задача F. ЗАПАЗЕНА МАРКА

След мощен запой с минерална вода, Станчо и Манчо сериозно се скарали. Станчо се обзаложил на 4 каси минерална вода с Манчо, че всяко число, което завършва на 4 се дели на 2. Лукавият Манчо, обаче, посочил за контрапример числото 3,24 и отмъкнал касите. Сега Станчо е накърнен и обеднял (защото, както знаете, минералната вода поскъпна). За това той е решен да си върне водата обратно, като атакува Манчо в съда.

Манчо има много фирми, но не е регистрирал имената им като запазени марки. След анализ на действащото законодателство, Станчо установил, че може да регистрира като запазени марки низове, които се съдържат в имената на фирмите на Манчо, при което Манчо ще трябва да му изплати компенсация. За да не изглежда нагъл, Станчо смята да атакува името α на всяка фирма на опонента с точно един подниз β . По закон, при такъв иск, Манчо трябва да изплати като компенсация левовата равностойност на $n = m \cdot l$ бутилки минерална вода от 500 ml, където m е броят на срещанията (може и застъпващи се) на β в α , а l – дължината на β .

Ако Манчо има фирма **alalabalala**, например, тогава Станчо би могъл да регистрира като запазена марка низа **a**, при което Манчо трябва да му изплати цената на 6 бутилки вода (6 срещания на низ с дължина 1 буква). Ако Станчо го атакува със запазената марка **ala**, обаче, Манчо ще трябва да му изплати цената на 12 бутилки (4 срещания на низ с дължина 3 букви). Станчо може да регистрира и цялата дума **alalabalala** (никой закон не е безгрешен), при което ще получи като компенсация цената на 11 бутилки. Напишете програма, която за всяка фирма на Манчо, пресмята цената на колко най-много бутилки вода може да получи Станчо.

Вход

На стандартния вход ще бъдат зададени имената на фирмите на Манчо. Всяко име е записано на отделен ред и представлява низ от най-много 2000 малки латински букви.

Изход

За всяко зададено на входа име на фирма, програмата трябва да изведе на отделен ред на стандартния изход едно число – максималния брой бутилки, цената на които Станчо може да получи от атака на името на тази фирма.

Пример

Вход	Изход
Alalabalala	12
blablablabla	18
stanchoeglupavood	17

Задачата е предложена от Милослав Средков

XX РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
СОФИЯ, 9–10 МАЙ 2008

Задача G . КОЛОНА

При разкопки в покрайнините на село Кюлевча, изследователите от местния археологически клуб “Каквото открием за нас си е!” се натъкнали на три вида каменни блокове, с които били строени древните тракийски колони. Първия вид били с височина един метър, вторият вид – с височина два метра, а третия вид – с височина три метра. Известен факт било, че при строенето на колони, древните траки никога не поставяли два блока с еднаква височина един върху друг. Изследователите започнали да се чудят, по колко различни начина древните траки могли да построят една колона с височина N метра, използвайки трите вида каменни блока и следвайки изискването да не се поставят два блока с еднаква височина един върху друг. Друг известен факт било, че изследователите от клуба изобщо не могли да решават такива задачи и оставят това на вас.

Напишете програма, която пресмята броя на всички възможни начина за подреждания на трите вида каменни блокове един върху друг, така че да се получи колона с височина N , никои два съседни блока на коятода не са от един и същ вид.

Вход

На първия ред на стандартния вход е зададено цялото число T – броят на тестовите примери. За всеки тестов пример на отделен ред е зададена височината N на колоната ($1 \leq N \leq 100$).

Изход

За всеки тестов пример, на отделен ред на стандартния изход програмата трябва да изведе броя на всички възможни подредби на трите вида каменни блока един върху друг така, че никои два съседни да не са от един и същ вид.

Пример

Вход	Изход
2	1
1	3
4	

Задачата е предложена от Бисерка Йовчева

XX РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
СОФИЯ, 9–10 МАЙ 2008

Задача Н. ТОЧКОМАНИЯ

Няколко приятели играят игра, която се нарича „Точкомания”. Във всеки рунд на играта, всеки играч избира една точка в равнината с целочислени координати. Резултатът на всеки един от играчите е броя на точките на противниците, които са с по-малка или равна абсциса и с по-малка или равна ордината. Играчите нямат право да избират точки, които вече са избрани от друг играч. Напишете програма, която, по зададени координати на избраните точки, намира резултата на всеки от играчите.

Вход

На първия ред на стандартния вход е зададено цяло положително число T – броят на тестовите примери. Всеки тестов пример започва с броя N на играчите ($1 \leq N \leq 50\,000$). На i -тия от следващите N реда са зададени координатите X_i и Y_i на i -тия играч ($1 \leq X, Y \leq 100\,000$).

Изход

За всеки тестов пример, на отделен ред на стандартния изход, програмата трябва да изведе резултата на един от играчите, като резултата на i -тия играч е на i -тия ред.

Пример

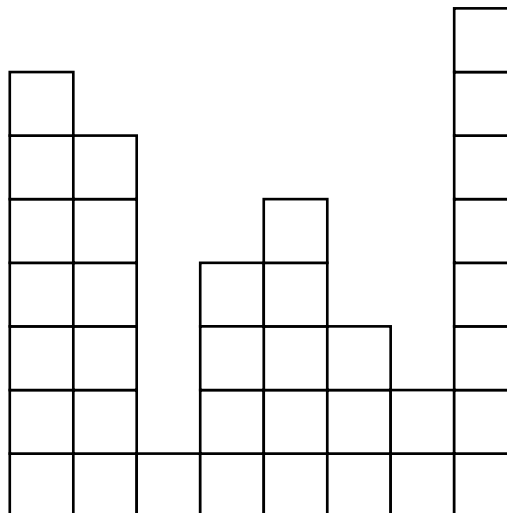
Вход	Изход
2	0
3	1
1 1	2
2 2	2
3 3	1
5	3
7 1	0
5 1	1
5 5	
1 1	
3 3	

Задачата е предложена от Петър Петров

XX РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
СОФИЯ, 9–10 МАЙ 2008

Задача I. ДЪЛГА РЕДИЦА ОТ КУЛИ

Интересът към построяване на високи сгради (кули) в познат ви морски курорт, дори при странните правила въведени от администрацията продължава да расте. Затова властите регулирали нови N еднакви парцела (сега N е значително по-голямо), разположени в редица край морския бряг и ги предложили на N строителни фирми. Всяка от фирмите бързо представила проект за построяване на съответната кула, като в проекта на всяка фирма бил заложен максималният брой етажи, които фирмата може да си позволи да построи. Оказало се, че в получената редица всичките N цели положителни числа са различни. Архитектът на курорта си имал своя идея за това как



трябва да изглеждат кулите и решил да избере един интервал от редицата и да разреши строителство само на тези фирми, парцелите на които попадат в избрания интервал, като на всички фирми разреши да построят по толкова етажа, колкото е минималният брой, който някоя от фирмите в интервала е заявила. Разбира се, добре е така да се подбере интервалът, че общия брой на построените етажи да е максималният възможен. Напишете програма, която да намери този максимален общ брой етажи.

Вход

На първия ред на стандартния вход ще бъде зададен броят T на тестовите примери. На първия ред за всеки тест ще бъде зададено цялото число N ($3 \leq N \leq 1\,000\,000$). На втория ред ще бъдат зададени N положителни цели числа, не по-големи от $1\,000\,000$ – броят на етажите които фирмите са заявили, подредени така както са подредени парцелите им на морския бряг.

Изход

На единствения ред на стандартния изход програмата трябва да изведе намерения максимален брой етажи.

Пример

Вход	Изход
1	12
8	
7 6 1 4 5 3 2 8	

Задачата е предложена от Красимир Манев

XX РЕПУБЛИКАНСКА СТУДЕНТСКА ОЛИМПИАДА ПО ПРОГРАМИРАНЕ
СОФИЯ, 9–10 МАЙ 2008

Задача J. ПРОИЗВЕДЕНИЕ НА ДВЕ ПРОСТИ ЧИСЛА

Днес едно от важните приложения на простите числа са шифрите за защита на информацията. При шифрите с публичен ключ, базирани на факторизацията на прости числа, публичният ключ е число M , което е произведение на две големи и различни прости числа, $M = AB$. Алгоритъмът за шифриране използва числото M , за да шифрира съобщението. Алгоритъмът за дешифриране изисква знанието на простите множители A и B . Така дешифрирането е лесно, ако са известни простите множители, които се съдържат в частния ключ и е изключително трудно, ако не са известни.

Напишете програма, която по зададено M да намира числата A и B .

Вход

На стандартния вход, разделени с интервал или край на ред, са зададени много цели положителни числа, всяко от които е по-малко от 2147483647.

Изход

За всяко число M от входа програмата трябва да отпечата на отделен ред на стандартния изход двете намерени числа A и B , като $A < B$.

Пример

Вход	Изход
15 221	3 5
196921	13 17
	191 1031

Задачата е предложена от Николай Киров