Група X

**МНОГОЪГЪЛНИК**

Даден ви е N-ъгълник (несамопресичащ се, може да е вдлъбнат). Напишете програма, която извежда броя на двойките вътрешни диагонали, които се пресичат във вътрешността на многоъгълника (контура не е част от вътрешността).Незащрихования диагонал е вътрешен, докато защрихованите не са.

**Input Format**

Първо се въвежда едно число T - брой тестове. За всеки тест ви се дава едно число , последвано от N реда с двойки цели числа, даващи координатите на поредната точка от многоъгълника.

**Constraints**

1 <= T <= 10

Всички координати са в интервала [-10^9, 10^9]

Тест 1 (10 точки): 3 <= N <= 1000 и многоъгълника е изпъкнал

Тест 2 (30 точки): 3 <= N <= 100

Тест 3 (60 точки): 3 <= N <= 1000

**Output Format**

За всеки тест изведете търсеното число на отделен ред.

**Sample Input 0**

1

9

3 2

3 0

5 -2

7 0

8 -2

10 0

8 4

6 2

5 4

**Sample Output 0**

27

**ВАФЛИ**

Еси и Дани са се озовали в склад за вафли. Вафлите били 2 вида – с кокос и с фъстъци. Разбрали се Дани да изяде вафлите с кокос, а Еси тези с фъстъци. След известно време ядене им доскучало и решили да доядат вафлите играейки следната игра:

Останали са N кутии с вафли, като във всяка кутия има и от двата вида. Двамата се редуват да ядат вафли, като играчът който е на ход избира една от кутиите и изяжда колкото иска вафли от нея (но само от неговия вид). Когато в някоя кутия останат само вафли от единия вид, играчът на който се полагат е длъжен да ги изяде на екс - това не се брои за ход. Играчът който не може да направи ход губи.

Еси е първа на ход. Вашата задача е да определите кой ще спечели играта ако и двамата играят оптимално.

**Input Format**

* На първия ред се въвежда числото T – броя тестове.
* За всеки тест първия ред е числото N – броя кутии. На всеки от следващите N реда има по две числа – броя вафли с кокос и броя вафли с фъстъци в съответната кутия.

**Constraints**

* 1 ≤ T ≤ 20
* 1 ≤ N ≤ 100 000
* 1 ≤ Броя вафли от един вид от една кутия ≤ 1018

**Output Format**

* Изведете T реда. На i-тия изведете “E” ако при оптимална игра от страна на двамата в i-тата игра печели Еси, иначе изведете “D”.

**Sample Input 0**

2

1

23 37

2

5 8

11 7

**Sample Output 0**

E

D

**Explanation 0**

* В първия пример Еси печели като изяде всички вафли с фъстъци от кутията. После Дани е длъжен да изяде останалите на екс (което не се брои за ход). Дани няма ход и Еси печели.
* В този случай печели Дани. Стратегията му е да взима по една вафла от по пълната кутия докато не остане 1 кутия. Тогава взима всички вафли и печели (аналогично на първия пример). Очевидно ще може да го направи, защото може да направи 14 хода преди да му се наложи да довърши вафлите си от някоя кутия, а Еси - само 13 (освен това тя е първа). Тоест тя ще довърши вафлите си от някоя кутия първа.

**ДОНЪТИ**

Не сте ли чували за световно известните донъти от Донътленд? За да бъдат толкова вкусни, те се приготвят от най-висококачествените продукти и съответно - най-скъпите. За да бъдат конкурентоспособни с Тортите на Директора, фабриките съкращават всякакви разходи, включително и заплатите на служителите си. Кралица Дени не е доволна от тази несправедливост в кралстовото си и затова предлага нов начин за разпределяне на парите в Кралската Фабрика за Донъти.

Във фабриката работят **N** служители, номерирани с естествените числа от 1 до **N** включително. Като изключим Дени (тя е служител с номер 1), всички работници имат точно един пряк шеф. Нека наричаме множеството от служители, чийто пряк шеф е даден друг служител, неговите преки подчинени. Валутата в Донътленд е Дени Долар (отбелязва се с **Đ**) и е дефинирана като точната стойност на един шоколадов донът. Тъй като никой не иска половин донът, всички банкноти са на стойност естествено число **Đ**. Според Дени, всеки служител трябва да разделя парите си по равно на своите подчинени. Поради факта, че не винаги тези пари се делят на броя на преките подчинени на съответния служител, той задържа за себе си остатъка от делението, а на подчинените си дава цялата част. Те на свой ред разпределят по същия начин парите си измежду своите преки подчинени и т.н. Ако работник няма преки подчинени, задържа всичките пари за себе си.

Макар и да харесва тази нова система, Дени разбира и че не винаги ще достигнат достатъчно **Đ** до всички служители. За да се справи с този проблем, Дени избира да дава на някои служители лично пари - средства, извън тези, получени от техния пряк шеф. Тези пари обаче отново се разпределят между преките подчинените на служителите по горната схема.

Дени ще ви даде структурата на йерархията на служителите във фабриката, като първоначално всички работници имат 0 **Đ**. След това тя ще ви даде **Q** на брои заявки, които да изпълните. Те могат да бъдат два вида:

1. Дени дава някаква сума **Đ** на определен служител, която се разпределя измежду подчинените му по горната схема
2. Дени ви пита колко пари има даден служител. Тази стойност, разбира се, не включва парите, които служителят е дал на преките си подчинени.

Важно е да се отбележи, че парите, които всеки работник разпределя измежду преките си подчинени при дадена заявка от тип 1, се състоят само от тези, получени от прекия му шеф при текущата заявка, и не включват парите, които са му останали от предходни заявки. Отговорът на заявка от тип 2 е сумата от парите, останали на служителя след всяка предходна заявка от тип 1.

**Input Format**

Първият ред на стандартния вход се състои от едно естествено число **N** - броя на служителите в Кралската Фабрика за Донъти. Следва втори ред с **N** - 1 числа, където **i**-тото число съответства на номера на прекия шеф на служител с номер **i** + 1. Третият ред съдържа естественото число **Q**, а следващите **Q** реда по една заявка. Заявките от тип 1 се дават във вида 1 **X Y**, където **X** е номерът на служителят, а **Y** броя **Đ**, които Дени му дава.

Заявките от тип 2 се дават във вида 2 **X**, където **X** е номерът на служителя, за който Дени пита.

**Constraints**

Във всички тестове 1 ≤ **N** ≤ 200 000 и 1 ≤ **Q** ≤ 1500, за всяка заявка 1 ≤ **X** ≤ **N** и 1 ≤ **Y** ≤ 109.

1. За 20% от точките 1 ≤ **N** ≤ 1500
2. За други 40% от точките 1 ≤ **Y** ≤ 100
3. За останалите 40% няма допълнителни ограничения

**Output Format**

За всяка заявка от тип 2 изведете на отделен ред едно цяло число - броя **Đ**, останали на съответния работник.

**Sample Input 0**

9

1 1 3 3 3 4 7 7

7

1 1 5

1 3 1

2 2

2 5

1 4 1

2 7

2 3

**Sample Output 0**

2

0

1

3