

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 април 2012 г.

Задача А. Окръжности

В равнината са дадени са n окръжности: C_1, C_2, \dots, C_n . Разглеждаме неориентиран граф с върхове дадените окръжности, като между два върха има ребро когато съответните окръжности имат точно две общи точки. Напишете програма, която намира броя на ребрата в най-късия път от C_1 до C_n .

На първия ред на **стандартния вход** е даден броя на тестовите примери. Всеки от тях започва с числото n ($2 \leq n \leq 1000$), а на следващите n реда са записани по три цели числа x, y и r , задаващи координатите на центъра и дължината на радиуса на поредната окръжност ($-10000 < x, y < 10000$; $0 < r < 10000$). Номерата на окръжностите (C_1, C_2, \dots, C_n) е според наредбата им във входния файл.

За всеки тестов пример на **стандартния изход** да се изведе търсения брой или -1 , ако не съществува път от C_1 до C_n .

Пример:

Вход	Изход
1 3 0 0 1 4 0 4 1 0 2	2

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 април 2012 г.

Задача В. Начупена линия

В равнината са дадени n точки. Търси се дължината на най-късата начупена линия, минаваща през всяка от дадените точки и съставена само от хоризонтални и вертикални отсечки.

На първия ред на **стандартния вход** е зададен броя на тестовите примери. Всеки от тях започва с числото n ($1 < n < 10$), а на следващите n реда са дадени по две цели числа x и y – координатите на поредната точка ($0 < x, y < 10$).

За всеки тестов пример на **стандартния изход** да се изведе дължината на най-късата начупена линия.

Пример:

Вход	Изход
1 4 1 2 2 6 5 2 2 2	9

Обяснение на примера: Минималната възможна дължина е 9. Линията $(2, 6) - (2, 2) - (1, 2) - (5, 2)$ е едно възможно решение, за което тази дължина се достига.

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 април 2012 г.

Задача С. Мравки и щурци

Мравешка колония е построила дълъг и прав тунел. В него всяка мравешка фамилия си е построила жилище, в което всички мравки от фамилията живеят и съхраняват хранителни запаси за зимата.

Тъй като мравките трупат много по-големи запаси отколкото са им нужни, Кралицата мравка решила всяка фамилия да покани по няколко щурци за Коледа (всяка фамилия трябва да покани един и същ брой щурци).

Известен е броят на излишните житни зърна, които всяка фамилия може да предостави на щурците, като на всеки щурец ще бъде дадено точно едно зърно. Житните зърна, които една фамилия дарява на щурците, могат да бъдат изядени от нейните гости или да бъдат дарени за гостите на друга фамилия. Във втория случай, те трябва да бъдат транспортирани. За пренасянето на житни зърна до друго жилище, мравките изяждат по едно зърно за всеки изминат метър в тунела, независимо колко зърна пренасят.

Напишете програма, която пресмята максималният брой щурци, които могат да гостуват на всяка мравешка фамилия.

От първия ред на **стандартния вход** се въвежда броя на тестовите примери. Всеки от тях започва с едно цяло число N – брой на мравешките фамилии. От следващите N реда се въвежда информация за всяка фамилия: ред $i+1$ съдържа две цели числа A_i и B_i , където A_i е позицията на жилището на мравешка фамилия i (представена като разстояние в метри от входа на тунела), а B_i е броят на излишните житни зърна на същата фамилия.

За всеки тест, на отделен ред, програмата трябва да изведе, едно цяло число – максималният брой щурци, които могат да гостуват на всяка мравешка фамилия.

Ограничения

$1 \leq N \leq 100000$, $1 \leq A_i \leq 1000000000$, $0 \leq B_i \leq 1000000000$, $i = 1, 2, \dots, N$. Няма две жилища, разположени на една и съща позиция.

Пример:

Вход	Изход	Пояснение към примера
1 4 20 300 40 400 340 700 360 600	415	Четири мравешки фамилии са заселени на 20, 40, 340 и 360 метра от входа на тунела. Всяка фамилия може да приеме 415 щурца. Фамилия 1 има 300 излишни зърна и получава още 115 от фамилия 2, които се транспортират на 20 метра. След това фамилия 2 ще разполага с $400 - 115 - 20 = 265$ житни зърна. Фамилия 2 има 265 зърна и получава още 150 от фамилия 3, които се транспортират на 300 м. Така фамилия 3 ще разполага с $700 - 300 - 150 = 250$ житни зърна. Фамилия 3 получава от фамилия 4 165 житни зърна, които се транспортират на 20 м, след което за фамилия 4 остават $600 - 20 - 165 = 415$ житни зърна. Така на всяка фамилия ще гостуват по 415 щурци и този брой е възможно най-големият.

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 април 2012 г.

Задача D. Машината на Бойко

Бойко Разбойков изобретил машина, която яде цели положителни числа и плюе костилките. За всяко цяло положително число n ($n \leq 10^{100}$), костилката му е едноцифрено число. Машината намира костилките по следния начин: От числото n се получава ново число n_1 , цифрите на което представляват абсолютните стойности на разликите на съседните цифри на числото n . Аналогично от числото n_1 се получава ново число n_2 и т.н. до достигане на костилката – едноцифреното число n_i . Напишете програма, която да намира костилката на въведено цяло положително число n .

Програмата трябва да обработи множество числа, зададени на отделни непразни редове на **стандартния вход**.

За всяко число на отделен ред на **стандартния изход** да се изведе по едно цяло число – костилката му, изплюта от машината на Бойко Разбойков.

Пример:

Вход	Изход
9021123	5

Пояснение

$n=9021123$ т.е. $n_1=|9-0| |0-2| |2-1| |1-1| |1-2| |2-3|$

$n_1=921011$

$n_2=71110$

$n_3=6001$

$n_4=601$

$n_5=61$

$n_6=5$ е костилката на числото 9021123.

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 април 2012 г.

Задача Е. Опашка

В повечето банкови институции, за по-доброто обслужване на клиентите, вече са инсталирани компютризирани системи. Една такава система регистрира пристигналия в банката клиент, подрежда го в опашка и, когато се появи възможност за обслужване, извиква от опашката клиента, който е наред да бъде обслужен. Затова и новосъздадената Първа Приоритетна Банка (ППБ) решила да внедри подобна система, като отчете някои свои особености. Клиентите на ППБ се идентифицират с постоянен клиентски номер k и при всяко свое идване в банката получават различен приоритет за обслужване p , с който се нареждат на опашката. Когато дойде време за обслужване, от опашката се извиква клиентът с най-голяма стойност на приоритета в момента. Като на всяка опашка, клиентите чакащи за обслужване нервничат. Не рядко някой от чакащите се обръща към обслужващия персонал и пита „Колко клиенти има преди мен на опашката в момента?“, а ако в резултат получи твърде голямо число, може да реши и да се откаже да чака повече. Като програмист на банката трябва да реализирате програмно основна част от бъдещата обслужваща система.

Програмата трябва да обработва заявки от следния вид:

- 1 $k p$ – клиентът с номер k се нарежда на опашката с приоритет p ;
- 2 – клиентът с най-голям приоритет се изважда от опашката;
- 3 k – клиентът с номер k пита колко души има преди него в опашката;
- 4 k – клиентът с номер k напуска опашката без да бъде обслужен,

като $1 \leq k \leq 10^6$, $1 \leq p \leq 10^7$. Заявките се четат от **стандартния вход**, като на всеки ред е записана по една заявка. Последният ред на входа съдържа само 0. Не е възможно в заявка от тип 1 да се появи номер или приоритет, които вече се намират в опашката, нито пък в заявки от тип 3 и 4 – номер на клиент който не е в опашката.

За всяка заявка от тип 2 програмата трябва да изведе на отделен ред на **стандартния изход** номера на клиента, който е изваден от опашката за обслужване, а ако опашката е празна – да изведе 0. За всяка заявка от тип 3 програмата трябва да изведе на отделен ред на стандартния изход търсения брой клиенти.

Пример:

Вход	Изход
1 5 1000	0
3 5	1
1 1 1002	1
3 5	6
1 6 300	
2	
4 5	
2	
0	

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 април 2012 г.

Задача F. Правоъгълник

Единият от върховете на правоъгълник съвпада с началото на координатната система, а други два от върховете му са точки с целочислени координати (x_1, y_1) и (x_2, y_2) . Напишете програма, която намира колко от зададени N на брой различни точки лежат във вътрешността или върху някоя от страните на правоъгълника.

На първия ред на **стандартния вход** ще бъде зададен броят на тестовете. На първия ред на всеки от тях са дадени 5 числа – координатите x_1, y_1, x_2 и y_2 на два от върховете на правоъгълника и броя N на зададените точки ($0 < N < 1000$). На следващите N реда са описани координатите на тези точки. Всички зададени координати са цели числа, които по абсолютна стойност не надминават 1000.

За всеки пример трябва да изведете на **стандартния изход** колко от зададените точки лежат във вътрешността или върху някоя от страните на правоъгълника.

Пример:

Вход	Изход
1 3 11 5 1 5 0 0 1 3 5 0 0 11 -2 11	2

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 април 2012 г.

Задача G. Изпъкнал многоъгълник

В равнината са дадени n различни точки с целочислени координати. Напишете програма, която определя дали е възможно точките да се подредят така, че да бъдат последователни върхове на изпъкнал n -ъгълник. Ако това е възможно да се пресметне лицето на многоъгълника.

На първия ред на **стандартния вход** ще бъде зададен броят на тестовете. На първия ред на всеки от тях е зададено числото n ($3 \leq n \leq 13$). След това за всяка точка на отделен ред са дадени нейните координати x, y ($0 \leq x, y \leq 20$).

За всеки пример трябва да изведете на **стандартния изход** едно число – лицето на изпъкнал n -ъгълник с върхове дадените n точки. Ако не съществува такъв многоъгълник, да се изведе числото 0.

Пример:

Вход	Изход
3	0
4	14.5
1 1	0
5 1	
10 1	
6 3	
4	
0 0	
4 0	
0 3	
3 5	
5	
0 0	
5 1	
3 3	
2 7	
6 5	

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 април 2012 г.

Задача Н. Игра

Сашо и Марто играят следната игра. На дъска с размери $N \times M$ са поставени k пула. Играчите се редуват, като всеки път Сашо започва, местейки един пул в някоя от съседните му клетки (две клетки са съседни, ако имат обща страна). Ако някой от пуловете се намира на границата на дъската, с един ход Сашо може да го извади от дъската и да спечели. След като Сашо е направил своя ход, Марто избира една външна страна на някое от квадратчетата на дъската и я маркира (външна е страна, която не се намира между две различни квадратчета). Никой пул не може да преминава маркирана страна на квадратче. Позволено е в едно и също квадратче на дъската да има повече от един пул.

Въпросът е: Има ли шанс Сашо да спечели при оптимална игра на Марто?

Вход

Програмата трябва да обработва t тестови примера. На първия ред е дадено числото t . На следващите редове са описани входните данни за всеки от тестовите примери. Входните данни за един тестов пример се задават в следния вид: на първия ред са записани числата N ($1 \leq N \leq 100$), M ($1 \leq M \leq 100$) и k ($0 \leq k \leq 100$). На всеки от следващите k реда са записани по две числа – позицията на текущия пул върху дъската.

Изход

За всеки тестов пример, в зависимост от отговора на въпроса, програмата трябва да изведе YES или NO на отделен ред.

Пример:

Вход	Изход
2	YES
2 2 1	NO
1 2	
20 20 2	
10 10	
9 9	

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 април 2012 г.

Задача I. ТВЪРДО „НЪЕ“ НА ФИБОНАЧИ

История:

Редицата от естествени числа 1, 1, 2, 3, 5, 8, ..., в която първите два елемента са единици, а всеки от останалите е сума от предишните два, носи (може би не съвсем заслужено) името на Леонардо Фибоначи от Пиза.

Задачата:

Всичко живо се буначи с тез числа на Фибоначи! Ние питаме, обаче: ами другите играчи? Как са наредени, значи, тия вечни изплъзвачи, защитени от това, че те **НЕ СА** на Фибоначи?

Казано по-просто:

Кое е n -тото естествено число, което не е на Фибоначи?

Вход

На първия ред на **стандартния вход** ще бъде зададен броят на тестовете. На първия ред на всеки от тях е зададено естественото число n – номер на „нефибоначиево“ число.

Изход

За всеки пример трябва да изведете на **стандартния изход** едно число – n -тото в редицата, която остава, като изтрием всички числа на Фибоначи от редицата на естествените числа $\{1, 2, 3, 4, \dots\}$. Броенето започва от 1.

ОГРАНИЧЕНИЯ: $n > 0$ и има не повече от 18 цифри.

Пример:

Вход	Изход
1 10	16

Обяснение на изхода:

Първите числа в редицата на не-Фибоначи са: 4, 6, 7, 9, 10, 11, 12, 14, 15, 16, 17, ...

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 април 2012 г.

Задача J. Ягоди

Иванчо и Петранчо играят игра. На масата има N на брой купи с ягоди, всяка от които съдържа $a[i]$ ягоди ($0 \leq i < N$). Като се редуват, на всеки ход един от тях взима произволен брой ягоди от избрана от него купа. Губи този, който не може да направи ход. Първи на ход е Петранчо, тъй като Иванчо е домакин и иска да му даде преднина. Но давайки му преднина, Иванчо иска да се подsigури предварително, че той ще спечели. Преди Петранчо да му дойде на гости, той може да сложи още X на брой купи с ягоди, като всяка от тях може да съдържа не повече от K ягоди. Помогнете на Иванчо като напишете програма, която да намери минималното X , за което той със сигурност ще спечели играта. Ако това е невъзможно, ще считаме, че $X = -1$.

Вход

Програмата трябва да обработва T тестови примера. На първия ред е зададено числото T . На следващите $2T$ реда са описани входните данни за всеки тестов пример. Входните данни за един тестов пример са описани на два реда. На първия ред са зададени целите положителни числа N и K . На следващия ред са зададени $a[0], a[1], \dots, a[N - 1]$.

Изход

За всеки тестов пример програмата трябва да изведе на отделен ред съответното число X .

Ограничения:

$$2 \leq N \leq 1000$$

$$1 \leq K \leq 1000$$

$$1 \leq a[i] \leq 1000$$

Пример:

Вход	Изход
2	2
3 9	0
2 7 10	
3 3	
2 6 4	

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 22 април 2012 г.

Задача К. Вероятности

Бай Пешо много обичал да си играе със зарове и монети. Обичал всякакви видове. От монети до 36 страни зара. Той много искал да разбере какъв е шансът да му се падне определено число, хвърляйки няколко еднакви зара. Като това число е сборът на заровете. Напишете програма, която по броя и вида на заровете да пресмята каква е вероятността сборът на заровете да е зададено число. Страните на заровете са номерирани с поредни числа, започващи с 1. Предполагаме се, че заровете са честни, т.е. вероятността да се падне всяка страна е една и съща.

Според теорията, вероятността да се случи едно събитие е равна на броя на благоприятните случаи, разделен на броя на всички възможни случаи.

На първия ред на **стандартния вход** ще бъде зададен броя на тестовете. За всеки тест на отделен ред ще бъдат зададени N , K и S , където N е броят на заровете, K е броят на страните на зара и S е сборът на числата, чиято вероятност търсим.

Ограничения: $1 < N < 16$, $2 < K \leq 16$ и $N < S \leq N * K$.

За всеки тест на отделен ред на **стандартния изход** програмата трябва да изведе вероятността в проценти, с точност 5 знака след десетичната точка и знака '%'. Ако вероятността е по-малка от 10^{-5} да се изведе '0%'.

Пример:

Вход	Изход
4	8.33333%
2 6 4	50.00000%
2 2 3	16.66667%
1 6 6	13.88889%
2 6 8	

Обяснение на първия пример: Има 3 благоприятни случаи (да се падне 4) при хвърляне на два 6-страни зара, а именно 1+3, 2+2 и 3+1. Всички възможни случаи са $6^2 = 36$ и за търсената вероятност получаваме $3/36$, което е приблизително 8.33333%.