

**Нов български университет**  
**Департамент Информатика**  
**Школа „Състезателно програмиране“**  
**СЪСТЕЗАНИЕ, 12 януари 2013 г.**

**Задача А. Допълване до квадрат**

Дадено е цялото положително число  $A$ . Напишете програма, която да намери най-малкото естествено число  $B$  такава, че  $AB$  е точен квадрат.

На първия ред на стандартния вход ще бъде зададен броят на тестовите примери, които програмата трябва да реши. Входът за всеки тестов пример се състои от един ред на който е зададено числото  $A$  ( $A < 10^9$ ).

За всеки тестов пример, програмата трябва да изведе на стандартния изход намереното число  $B$ .

**Пример:**

Вход	Изход
2	2
2	6
17496	

**Нов български университет**  
**Департамент Информатика**  
**Школа „Състезателно програмиране“**  
**СЪСТЕЗАНИЕ, 12 януари 2013 г.**

**Задача В. Телефони**

През лятото, плажът на морския курорт „Гергана“ е претъпкан с клиенти на мобилния оператор Гол-Тел. За да могат динамично да управляват трафика, програмистите на компанията трябва да напишат съответен софтуер. Те разделили плажа на  $N$  зони, в центъра на всяка от които е един от постове на плажните спасители, номерирани от 1 до  $N$ . В началото на работата на програмата, във всяка от зоните има определен брой клиенти на оператора –  $A_1, A_2, \dots, A_N$ . От време навреме в някоя от зоните се появяват нови клиенти или пък някои, които са били в зоната напускат. За да установи какво е натоварването в определени участъци на плажа, програмата трябва да може, при нужда, да дава отговор на въпроси от вида „Колко телефона работят в момента в зоните от пост  $X$  до пост  $Y$ , включително“. Напишете нужната програма.

**Вход**

На първия ред на стандартния вход ще бъде зададен броят на тестовете. Всеки тест започва с ред, който съдържа  $0 \ N \ A_1 \ A_2 \dots \ A_N$ ,  $N \leq 100000$ ,  $0 \leq A_i \leq 255$ . Следват поредица от заявки от един от двата вида:

а)  $1 \ Z \ C$  е указание, че в зона  $Z$  броят на клиентите се е променил с  $C$ ,  $-10 \leq C \leq 10$ . Ако  $C$  е отрицателно, гарантирано е, че броят на клиентите в зоната  $Z$  в този момент е поне  $-C$ .

б)  $2 \ X \ Y$  е въпрос за броя на клиентите в зоните от  $X$  до  $Y$ ,  $1 \leq X \leq Y \leq N$ .

Входните данни за теста завършват с ред на който е зададено само числото 3. Общият брой заявки не надхвърля 100000.

**Изход**

За всяка заявка от тип 2 програмата трябва да изведе на стандартния изход броя на действащите телефони в указания интервал.

**Пример:**

Вход	Изход
1	5
0 5 2 3 0 6 5	14
2 1 3	6
2 2 5	16
2 4 4	
1 3 2	
1 4 -2	
2 1 5	
3	

**Нов български университет**  
**Департамент Информатика**  
**Школа „Състезателно програмиране“**  
**СЪСТЕЗАНИЕ, 12 януари 2013 г.**

**Задача С. Подредица**

Дадена е редица  $X$  от  $N$  цели числа. Да се намери най-дългата подредица, сумата от елементите на която се дели (без остатък) на зададено число  $K$ . Казваме, че  $Z$  е подредица на  $X$ , ако  $Z$  може да бъде получена чрез премахване на някои членове на  $X$ . Празното множество е подредица (с дължина 0) и цялата редица също е подредица (с дължина  $N$ ).

**Вход**

Първият ред на входа съдържа две числа  $N$  и  $K$ . Следващите редове съдържат точно  $N$  цели числа - елементите на редицата. Входът съдържа повече от един пример и завършва с  $N = 0$ . Всички числа на входа са цели положителни числа, по-малки от 1000, с изключение на последното число, което е 0.

**Изход**

Всеки ред на изхода съдържа дължината на най-дългата намерена подредица за поредния пример от входа.

**Пример:**

Вход	Изход
7 5	5
1 1 1 2 2 2 4	9
10 3	0
1 2 3 4 5 6 7 8 9 10	6
5 27	
20 1 1 1 1	
6 6	
10 10 10 2 2 2	
0	

**Нов български университет**  
**Департамент Информатика**  
**Школа „Състезателно програмиране“**  
**СЪСТЕЗАНИЕ, 12 януари 2013 г.**

**Задача D. Лифт**

Фирма „Иди ми, дойди ми“ С някой връзки успя да уреди строежа на новия си лифт, свързващ световноизвестните ски-писти Горно Нанадолнище и Долно Нанагорнище. Поради финансови затруднения лифтът засега разполага само с една кабинка, но от фирмата са решили да го пуснат в експлоатация. В „Иди ми, дойди ми“ много разчитат на груповите пътувания и подготвят атрактивна оферта за своите клиенти:

- Цената за едно пътуване на турист между двете писти се определя от височината му по правилото – всеки плаща колкото е висок. Така, пътник висок 175 см, трябва да плати 175 евроцента.
- Ако няколко туристи пътуват едновременно, те заплащат заедно само цената, която трябва да плати най-високият от тях.

От фирмата имат и някои малки претенции:

- Теглото на пътниците в едно пътуване да не превишава  $T$  килограма.
- Лифтът да не пътува празен в никоя от двете посоки.

Група от  $N$  туристи иска да използва лифта за придвижване от Горно Нанадолнище до Долно Нанагорнище, но не знае колко ще трябва да плати. Съставете програма, която определя минималната цена, която групата трябва да плати.

**Вход**

От първия ред на стандартния вход се въвежда броят на тестовете. Всеки тест започва с ред, на който са зададени  $N$  и  $T$  ( $0 < N < 15$ ,  $100 < T < 500$ ). На всеки от следващите  $N$  реда са зададени височината в сантиметри и теглото в килограми на един от туристите. Всеки турист е висок най-много 200 см и тежи най-много 150 кг.

**Изход**

На единствения ред на стандартния изход програмата трябва да изведе минималната цена в евроцентове, която групата трябва да плати или 0, ако групата не може да бъде превозена.

**Пример:**

Вход	Изход
1 3 200 170 90 160 80 150 100	480

**Нов български университет**  
**Департамент Информатика**  
**Школа „Състезателно програмиране“**  
**СЪСТЕЗАНИЕ, 12 януари 2013 г.**

**Задача Е. Бързо придвижване**

В големите градове бързината на придвижването не винаги зависи само от дължината на маршрута, който трябва да бъде изминат, а също и от натовареността на движението и пропускливостта на отделните улици. Пропускливостта се определя от ширината на улицата, от качеството на настилката, от нейната безопасност и т.н. и, в края на краищата, се измерва с някакво цяло положително число, което можем да наричаме *ширина*. Колкото по-голяма е ширината, толкова по-голяма е пропускливостта на улицата. Ширината на един маршрут в града, обаче, е равна на ширината на най-тесната улица по маршрута. Напишете програма, която в зададен град с  $N$  кръстовища (номерирани с числата от 1 до  $N$ ) и  $M$  улици, всяка от които свързва две от кръстовищата, намира маршрут с най-голяма ширина от кръстовището с номер 1 до кръстовището с номер  $N$ , ако такъв маршрут съществува.

На първия ред на **стандартния вход** ще бъде зададен броят на тестовите примери. Всеки тест започва с ред, на който са зададени, разделени с един интервал, числата  $N$  и  $M$ ,  $2 \leq N \leq 1000$ ,  $1 \leq M \leq 100000$ . Всеки от следващите  $M$  реда ще съдържа по три цели положителни числа  $I, J$  и  $W$ ,  $1 \leq I \neq J \leq N$ ,  $1 \leq W \leq 30000$ , разделени с по един интервал, които означават, че кръстовищата с номера  $I$  и  $J$  са свързани с улица с ширина  $W$ .

За всеки тестов пример програмата трябва да изведе на отделен ред на **стандартния изход** най-голямата ширина на маршрут от кръстовището с номер 1 до кръстовището с номер  $N$ , при условие, че такива маршрути съществуват. В противен случай програмата трябва да изведе на стандартния изход числото 0.

Пример.

Вход	Изход
2	3
5 6	0
1 2 1	
1 3 3	
1 4 9	
2 5 10	
3 5 4	
4 5 2	
6 5	
1 3 4	
1 4 3	
3 4 5	
2 6 6	
5 6 7	

**Нов български университет**  
**Департамент Информатика**  
**Школа „Състезателно програмиране“**  
**СЪСТЕЗАНИЕ, 12 януари 2013 г.**

**Задача F. Война**

Държава имала  $N$  ( $2 \leq N \leq 1\,000\,000$ ) града, номерирани с числата от 1 до  $N$ , като между някои от тези градове съществували директни (не преминаващи през други градове) двупосочни пътища. Такива двойки градове ще наричаме съседни. Възможно било пътуването от всеки град до всеки друг град (евентуално преминавайки през други градове), но тъй като държава била доста бедна, това можело да става по единствен начин. Градовете, които имат само един съседен друг град наричаме „гранични“.

Надвисвала военна опасност и военното командване замислило да изгради команден център в някой от градовете. При това, главнокомандващият, поставил задача така да се избере градът за команден център, че разликата между дължината на пътуването от него до най-отдалечения граничен град и дължината на пътуването от него до най-близкия граничен град да е минимална. Ако градът с командния център е граничен, то се гледат разстоянията само до други, различни от него, гранични градове. Напишете програма, намира най-подходящия град, в който да изградят своя команден център.

За всеки тестов пример на първия ред на **стандартния вход** е зададено естественото число  $N$ . На всеки от следващите  $N - 1$  реда са зададени по три числа:  $x$ ,  $y$  и  $l$ . Всеки ред описва, че съществува път между градовете  $x$  и  $y$ , който е с дължина  $l$ .

За всеки пример на отделен ред на **стандартния изход** програмата ви трябва да изведе двойка числа, разделени с един интервал: номера на града, в който военните трябва да разположат своя команден център и намерената минимална разлика. Ако съществуват няколко решения, изведете това с най-малък номер на града за команден център.

Всеки директен път между два града е с дължина положително цяло число, не по-голямо от 1000.

Вход	Изход
6	1 4
6 2 2	
6 1 3	
6 4 2	
6 3 6	
1 5 7	

**Нов български университет**  
**Департамент Информатика**  
**Школа „Състезателно програмиране“**  
**СЪСТЕЗАНИЕ, 12 януари 2013 г.**

**Задача G. Рибите и орела Тошко**

За рибите в река Бистра настъпиха тежки времена. Орелът-рибар Тошко си направи гнездо наблизко и рибите вече не можеха да плуват безгрижно надолу-нагоре по реката, поне докато е светло. Щом се появи Тошко в небето, всяка риба се стрелкаше към най-близкото скривалище – някой подмол или под сянката на някое дърво. Реката беше намаляла през горещото лято, скривалищата останаха малки и често Тошко грабваше рибата, която не успяваше да се намести в скривалището.

За простота нека си представим реката като права линия. Рибите ще бъдат представени като  $N$  точки, а скривалищата – като  $M$  точки с целочислени координати върху нея. Възможно е няколко риби, няколко скривалища или няколко риби и няколко скривалища да имат едни и същи координати.

Предполагаме, че рибите изминават по единица разстояние за една секунда. Например ако някоя от тях е с координата 42 и иска да отиде в скривалища с координата 13, то за целта ще са й нужни 29 секунди. Ако пък скривалището беше на позиция 53, на рибата щяха да са нужни 11 секунди. Напишете програма, която изчислява за колко най-малко време всички риби могат да се скрият от погледа на орела Тошко.

**Вход**

На първия ред на стандартния вход е зададено едно число – броя на примерите. За **всеки пример** на първия ред са зададени целите числа  $N$ ,  $M$  и  $K$  – съответно броя риби, броя скривалища и максималния брой риби, които могат да се поберат в едно скривалище. На следващия ред има  $N$  цели числа  $S_1, S_2, \dots, S_N$  – координатите на рибите по правата. На третия ред са зададени  $M$  цели числа  $B_1, B_2, \dots, B_M$  – координатите на скривалищата по правата. Знае се, че  $1 \leq N, M, K \leq 100000$  и  $1 \leq S_i, B_i \leq 1000000$ .

**Изход**

За всеки пример на отделен ред на стандартния изход да се изведе едно цяло число – минималното необходимо време, за което всички риби могат да стигнат до скривалища, без скривалищата да се препълват (тоест всяко от тях да съдържа по  $K$  или по-малко риби). Ако това е невъзможно, изведете  $-1$ .

**Пример:**

Вход	Изход
1 7 3 3 4 9 8 2 4 6 5 2 7 2	3

Рибите с координати 4, 2, 4 и 5 се разпределят в двете скривалища с координати 2, а останалите – в скривалището с координати 7. Най-много време отнема на рибата с координати 5 да стигне до скривалището с координати 2.

**Нов български университет**  
**Департамент Информатика**  
**Школа „Състезателно програмиране“**  
**СЪСТЕЗАНИЕ, 12 януари 2013 г.**

**Задача Н. Сума**

Дадени са цели положителни числа  $A, B, P, Q$ . Напишете програма `sum`, която намира сумата на целите числа, които са не по-малки от  $A$ , не по-големи от  $B$  и са едновременно кратни на  $P$  и на  $Q$ .

**Вход**

За всеки тест, от един ред на стандартния вход се въвеждат целите положителни числа  $A, B, P, Q$ , разделени с по един интервал.

**Изход**

На един ред на стандартния изход се извежда намерената сума за всеки тест. Ако в интервала от  $A$  до  $B$  няма числа, които се делят едновременно на  $P$  и  $Q$ , да се изведе  $0$ .

**Ограничения:**

$$2 < P, Q < 100$$

$$0 < A < B$$

Вход	Изход
3 50 6 4	120
1 100 3 11	198



**Нов български университет**  
**Департамент Информатика**  
**Школа „Състезателно програмиране“**  
**СЪСТЕЗАНИЕ, 12 януари 2013 г.**

**Задача I. Големия Джо – фермер**

След като приключил с разбойничеството, защото все по-често шерифите започнали да хващат и бесят разбойниците, и "спечелил" от тази дейност някакви пари, Големия Джо решил да стане фермер, като повечето новодошли европейци в обширните земи на Америка. Отишъл в незаетите още равнини на средния запад, и като "порядъчен" нов заселник се представил на местния шериф:

- Искам земя, ще правя ранчо.

Шерифът обяснил накратко правилата:

- Отиваш в склада, купуваш колци и оградна мрежа. Колкото земя заградиш – твоя е! Така и направил нашия приятел – отишъл в склада, там останали 100 колци и 1000 метра мрежа! Колко мрежа и колко колци да купи? В равнината – празно място много, но колко и какви материали да купи с наличните пари, че да си загради най-голям участък? Помогнете на Джо!

**Вход**

На стандартния вход се задават много примери, всеки пример се задава с три числа - парите, с които Джо разполага, цената на един колец и цената на един метър мрежа (мрежата се продава с точност до метър, т.е. можеш да купиш само цяло число метри). Числата са с два знака след десетичната точка и са по-малки от 1000001.

**Изход**

За всеки пример на стандартния изход на отделен ред да се отпечатат закупените броя колци и метри оградна мрежа.

**Пример:**

Вход	Изход
100.00 1.50 0.60	7 149
20.00 5.00 2.00	3 2

**Нов български университет**  
**Департамент Информатика**  
Школа „Състезателно програмиране“  
СЪСТЕЗАНИЕ, 12 януари 2013 г.

**Задача J. Максимална сума**

Дадена е матрицата  $a$  от цели числа с размер  $N \times N$ . Имаме пул, който отначало се намира в клетка  $(1, 1)$  на тази матрица. На всеки ход можем да местим този пул надолу или надясно. Това ще рече, че ако пултът е в клетка  $(i, j)$  можем да го преместим в  $(i+1, j)$  или в  $(i, j+1)$ . Също така, можем да местим пула **най-много**  $K$  на брой пъти наляво или нагоре. Това ще рече, че можем да преместим най-много  $K$  на брой пъти пула (ако той стои в  $(i, j)$ ) в клетки  $(i-1, j)$  или  $(i, j-1)$ . В нито един момент пултът не може да бъде преместен извън дъската. Всеки път когато пултът стъпва върху някоя клетка от матрицата, числото в тази клетка се добавя към текуща сума. Разрешено е пултът да стъпва повече от веднъж върху една и съща клетка, като при това, при всяко стъпване, числото в тази клетка се добавя към сумата. Напишете програма, която намира най-голямата възможна сума при придвижване на пула по указаните правила, стартирайки от клетка  $(1, 1)$  и завършвайки в  $(N, N)$ , движейки пула по гореописания начин и използвайки най-много  $K$  стъпки наляво и нагоре.

**Вход**

На първия ред на стандартния вход са зададени естествените числа  $N$  и  $K$ . На следващите  $N$  реда са зададени по  $N$  цели числа, които образуват матрицата  $a$ . Входът съдържа много примери.

**Изход**

На един ред на стандартния изход програмата трябва да изведе максимална сума, която може да се постигне за съответния пример.

**Ограничения:**

$2 \leq N \leq 1000$ , всеки елемент на матрицата  $a$  е цяло число в интервала  $[-1000, 1000]$ ,  
 $0 \leq K \leq 100$ .

**Пример:**

Вход	Изход
3 1 1 1 0 1 1 0 1 1 1 4 4 1 1 1 0 1 0 1 0 1 1 1 0 0 0 1 1	7 15