



**Департамент Информатика**  
**Школа „Състезателно програмиране”**  
**СЪСТЕЗАНИЕ, 28 март 2015 г.**

**А. Пръчка**

Върху дълга пръчка са нанесени деления през всеки сантиметър. Началото на пръчката е с деление 0 и следват деления, означаващи 1 см, 2 см, ..., и т.н. до последното деление за  $n$  см в края на пръчката. Пръчката е направена от нееднороден материал и частта между деление 0 и деление 1 тежи  $a_0$  грама, частта между деление 1 и деление 2 тежи  $a_1$  грама и т.н. до частта между деление  $n-1$  и  $n$ , която тежи  $a_{n-1}$  грама. Пръчката е поставена в специална машина и работник може да избере деление от пръчката, така че машината да я разреже на две части по това деление. Частите падат на пода. Ако някоя част е дълга 1 см, работникът я взема и поставя в контейнера за готова продукция. Ако има части, по-дълги от 1 см, работникът взема такава част и я поставя в машината, за да я разреже. Този процес продължава, докато има части за разрязване. По време на процеса работникът постоянно вдига части. Напишете програма, която пресмята колко най-малко може да бъде сумарното тегло на повдигнатите части от работника.

**Вход**

На първия ред на стандартния вход е даден броят на тестовите примери. Първия ред на всеки от тях съдържа цялото положително число  $n$ . Стойностите  $a_0, a_1, \dots, a_{n-1}$  са дадени на втория ред и са разделени с интервали.

**Изход**

За всеки тест на отделен ред на стандартния изход да се изведе едно цяло число, равно на търсеното минимално тегло.

**Ограничения:**  $1 < n < 500$ ;  $0 < a_i < 100$ , за  $i = 0, \dots, n - 1$ .

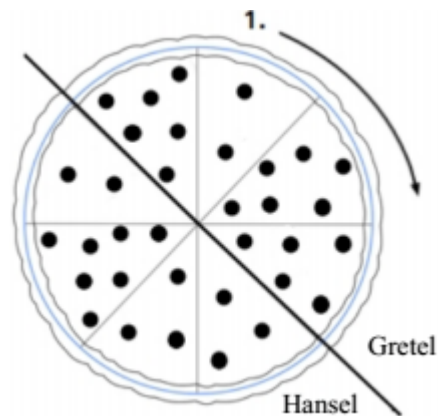
Вход	Изход
1	17
3	
1 6 3	

**Пояснение:** Разрязваме между парчетата, които са с тегло 6 и 3. Вземаме ги (едното – за да го сложим в машината, а другото – в контейнера) и така повдигаме  $(1 + 6) + 3 = 10$  грама. След това парчето в машината се разрязва на две парчета с дължина по 1 см и с тегла, съответно 1 и 6 грама. Така сумарно повдигаме  $10 + 6 + 1 = 17$  грама. Друг начин на разрязване води до по-голямо сумарно тегло на парчетата, които повдигаме.

**Департамент Информатика**  
**Школа „Състезателно програмиране”**  
**СЪСТЕЗАНИЕ, 28 март 2015 г.**

**В. Пица**

Хензел и Гретел много обичат да хапват пица, а Гретел най-много от всичко обича маслините в пицата. Двамата всеки път са изправени пред един и същи проблем... По дадени  $n$  ( $1 < n \leq 1000$ ) еднакви парчета пица и броят на маслините, във всяко едно от тях, те искат да подберат  $k$  ( $1 \leq k < n$ ) от тях, такива че сумата на маслините да е максималната възможна. Освен това подбраните парчета трябва да са едно до друго. Тези парчета ще получи Гретел, а останалите ще са за Хензел. Следния пример илюстрира, оптималното разпределение на парчета (1. обозначава първото парче от входния пример, а останалите са подредени по посока на часовниковата стрелка). Линията показва кой кои парчета ще получи.



Първия ред на всеки тестов пример ще съдържа целите положителни числа  $n$  и  $k$ . Следват  $n$  реда, като всеки от тях съдържа броя на маслините в поредното парче (този брой е цяло положително число в интервала  $[0; 100]$ ). Парчетата са зададени по посока на часовниковата стрелка, от първото към последното.

За всеки тест на отделен ред на стандартния изход да се изведе едно цяло число, равно на максималния брой маслини, които би получила Гретел.

Вход	Изход
8 4	19
2	
6	
5	
3	
3	
7	
2	
6	



**Департамент Информатика**  
**Школа „Състезателно програмиране”**  
**СЪСТЕЗАНИЕ, 28 март 2015 г.**

**С. Напоителна система**

Напоителната система се състои от един централен възел (свързан с кладенец, от където се черпи водата) и множество други възли, всеки от които може да бъде разпределител или пръскачка. Всички възли са свързани с тръби. Във всеки от възлите влиза точно една тръба, но може да излизат различен брой такива. Възлите от които не излизат тръби са пръскачките. На всяка от тръбите има монтиран кран, с който може да се спре водата, минаваща през него. Времето, което отнема за затварянето (съответно отварянето) на всеки един такъв кран е различно. По-новите се завъртат по-лесно, докато по-старите и ръждясали кранове изискват повече усилие. В зависимост от това време е съпоставено по едно число на всеки кран.

Да се напише програма, която минимизира времето за затваряне на нужните кранове, така че до пръскачките да не достигне вода.

Всеки тестов пример започва с броят на възлите  $n$  ( $1 < n \leq 1000$ ) и номера на централния възел  $c$  ( $1 \leq c \leq n$ ). Всеки от следващите  $n - 1$  реда описва една тръба и съдържа числата  $u$ ,  $v$  ( $1 \leq u, v \leq n$ ) и  $w$  ( $1 \leq w \leq 1000$ ), където  $u$  и  $v$  са възлите свързани с тръба, а  $w$  е усилието нужно за затваряне на крана, монтиран на тръбата.

За всеки тестов пример на нов ред извеждайте минималното време, нужно за прекъсване на водоподаването от централния кран до всички пръскачки.

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



**Департамент Информатика**  
**Школа „Състезателно програмиране”**  
**СЪСТЕЗАНИЕ, 28 март 2015 г.**

**D. Transposition cipher**

В криптографията Transposition cipher е метод за криптиране. Метода използва *ключ*, например думата **SECRET**. Първоначално всички повтарящите се букви от ключа се редуцират до една. Всички букви от латинската азбука се подреждат в таблица, като първоначално се записва ново получения ключ, след което се записват останалите букви от азбуката, като се пропускат тези от ключа. Таблицата има толкова колони, колкото са буквите от ключа.

```
SECRET
ABDFG
HIJKL
MNOPQ
UVWXY
Z
```

След което ключа се сортира, т.е. от **SECRET** се получава **CERST** и прочитаем колоните, започващи с поредната буква от сортирания ключ. По този начин на всяка буква от оригиналната азбука се съпоставя някаква друга:

```
Original:   ABCDE FGHIJ KLMNO PQRSTU VWXYZ
Substitution: CDJOW EBINV RFKPX SANMUZ TGLQY
```

Задачата ви е по зададен ключ, не по-дълъг от 7 символа и съобщение, състоящо се само от главни латински букви и празни символи, да намерите оригиналното съобщение. Празните символи си остават на мястото и не променени.

Първия ред на входа ще съдържа броя на тестовите примери. Всеки от тях се състои от два реда. На първия ще е зададен ключа, а на втория – криптирано съобщение, състоящо се от не повече от 255 символа (главни латински букви). Ключа и съобщението ще съдържат поне по един символ.

Резултата за всеки тестов пример извеждайте на отделен ред на стандартния изход.

Вход	Изход
2 SPORT LDXTW KXDTL NBSFX BFOII LNBHG ODDWN BWK SECRET JHQSU XFXBQ	ILOVE SOLVI NGPRO GRAMM INGCH ALLEN GES CRYPT OLOGY



Департамент Информатика  
Школа „Състезателно програмиране“  
СЪСТЕЗАНИЕ, 28 март 2015 г.

**Е. Щастливи прости числа**

Щастливо просто число наричаме просто число, на което можем да редуцираме сумата от квадратите на цифрите му итеративно, докато не са получи 1. За целите на задачата 1 не е такова число, а 7 е най-малкото такова, защото:

$$\begin{aligned}7 &\rightarrow 7^2 = 49 \\49 &\rightarrow 4^2 + 9^2 = 97 \\97 &\rightarrow 9^2 + 7^2 = 130 \\130 &\rightarrow 1^2 + 3^2 = 10 \\10 &\rightarrow 1^2 + 0^2 = 1\end{aligned}$$

Задачата ви е да тествате зададени числа дали са щастливо прости.

На първия ред на стандартния вход ще бъде зададен броят на тестовите примери  $T$  ( $1 \leq T \leq 10000$ ). Всеки от следващите  $T$  реда ще съдържа идентификатора  $K$  на поредното число, както и самото число  $m$  ( $1 \leq m \leq 10000$ ), което трябва да бъде тествано.

За всеки тестов пример извеждайте на отделен ред на стандартния изход  $K$ ,  $m$  и **YES** или **NO**, в зависимост от това дали  $m$  е щастливо просто число или не, разделени с интервал.

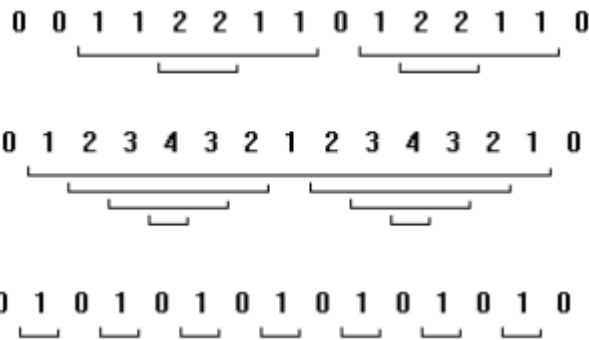
Вход	Изход
4	1 1 NO
1 1	2 7 YES
2 7	3 383 YES
3 383	4 1000 NO
4 1000	



Департамент Информатика  
Школа „Състезателно програмиране“  
СЪСТЕЗАНИЕ, 28 март 2015 г.

Ф. Острови

Дадена е редица от цели положителни числа  $a_1, a_2, \dots, a_n$ . *Остров* в редицата наричаме подредица от съседни елементи, всеки от които е по-голям от елементите непосредствено преди и след подредицата. Следните три примера илюстрират идеята:



Напишете програма, която по зададена редица намира всички острови в нея.

Всеки тестов пример се състои от една редица, от  $n$  ( $2 < n \leq 100$ ) числа, всяко от които е не по-голямо от 100. Всяка редица ще започва и завършва с 0. Всяко следващо число ще се различава от предишното най-много с 1.

За всеки тестов пример извеждайте на отделен ред на стандартния търсения брой острови.

Вход	Изход
0 0 1 1 2 2 1 1 0 1 2 2 1 1 0	4
0 1 2 3 4 3 2 1 2 3 4 3 2 1 0	7
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	7
0 1 0	1



Департамент Информатика  
Школа „Състезателно програмиране“  
СЪСТЕЗАНИЕ, 28 март 2015 г.

**Г. Цифри**

Напишете програма, която въвежда естествено число  $k$  ( $0 \leq k \leq 9$ ) и естествено число  $n$  ( $0 < n \leq 10^{18}$ ) и извежда колко пъти се използва цифрата  $k$  за изписване на числата от  $0$  до  $n$ .

**Вход**

За всеки тестов пример програмата въвежда от стандартния вход две цели неотрицателни числа  $k$  и  $n$ , отделени с един интервал.

**Изход**

За всеки тестов пример програмата извежда на стандартния изход едно цяло неотрицателно число – отговора на задачата.

Вход	Изход
3 42	14



**Департамент Информатика**  
**Школа „Състезателно програмиране“**  
**СЪСТЕЗАНИЕ, 28 март 2015 г.**

**Н. Дърводобив**

***N***-те ( $1 \leq N \leq 1\,000\,000$ ) дърветата около двора на Панко са подредени в една редица и образуват нещо като естествена ограда. За всяко от тези дървета е известна височината му в метри – цяло положително число не по-голямо от  $1\,000\,000\,000$ . Панко е превърнал своя мотокар в нещо като режеща машина. Супермотокара му разполага с моторна резачка, прикрепена в предната му част, която може да се настройва да реже дървета на определена височина ***H***. Панко просто припалва суперчудото и се вози, а докато се вози, добива и дървесина. Така например ако ***H*** е 15, а дърветата са с височини 20, 15, 10 и 17, то след един пас дърветата ще останат със следните височини 15, 15, 10 и 15. Отрязаната дървесина е общо 7 метра (5 от първото и 2 от последното дърво), нека я означим с ***M***.

Напишете програма, която пресмята каква ще бъде максималната целочислена стойност на ***H***, която позволява да се добият поне ***M*** ( $1 \leq M \leq 2\,000\,000\,000$ ) метра дървесина.

Първия ред за всеки тестов пример съдържа целите числа ***N*** и ***M***. Следват ***N*** реда – височините на дърветата (в метри). Сумата от всички височини ще надвишава ***M***, като по този начин Панко винаги ще бъде в състояние да добие необходимите метри дървесина. Край на входа е -1.

Резултата извеждайте на отделен ред на стандартния изход.

Вход	Изход
4 7	15
20 15 10 17	36
5 20	
4 42 40 26 46	
-1	





Департамент Информатика  
Школа „Състезателно програмиране“  
СЪСТЕЗАНИЕ, 28 март 2015 г.

### I. Железница

Преди много години в една държава имало директна железопътна линия между всеки два града. Цената за придвижване от един град до друг била  $B$  парични единици. В днешни дни някои от тези линии са заменени с нови, както цената на пътуването по тях вече струва  $A$  парични единици. Така всеки два града продължават да бъдат свързани. Влакове пътуват от град  $X$  до град  $Y$  и в обратната посока на една и съща цена.

Напишете програма, което намира най-ниската цена на която може да се стигне от град 1 до град  $N$  без значение от броя на прекачванията.

Всеки тестов пример започва с  $N, K, A$  и  $B$  ( $2 \leq N \leq 500000$ ,  $0 \leq K \leq 500000$ ,  $1 \leq A, B \leq 500000$ ). Това са съответно броят на градовете, броят на новите линии, новата и старата цена на пътуването от град до град. Следват  $K$  реда съдържащи числата  $u_i$  и  $v_i$  ( $1 \leq u_i, v_i \leq N$ ,  $u_i \neq v_i$ ), които показват, че линията между градовете  $u_i$  и  $v_i$  е подменена. Всяка двойка градове ще се среща не повече от веднъж във входа.

За всеки тестов пример извеждайте на отделен ред търсената минимална цена.

Вход	Изход
5 4 1 4 1 2 2 3 2 4 3 5	3



Департамент Информатика  
Школа „Състезателно програмиране”  
СЪСТЕЗАНИЕ, 28 март 2015 г.

### Ж. Матрица

Дадено ви е:

- Матрица от цели положителни числа, на която редовете и колоните са сортирани в нарастващ ред, т.е.  $A[i, j] \geq A[i - 1, j]$  и  $A[i, j] \geq A[i, j - 1]$  за всяко  $i, j$ .
- Една или повече двойки  $(X, Y)$ , за които  $Y \geq X$ .

За всяка двойка  $(X, Y)$  да се определи колко са числата от матрицата по-големи или равни на  $X$ , но по-малки или равни на  $Y$ .

#### Вход

Данните (32-битови цели положителни числа) ще бъдат записани в **двоичен (бинарен) файл**. Файла се състои от:

- Цяло число  $N$ , указващо броят на редовете на матрицата (не повече от 10000).
- Цяло число  $M$ , указващо броят на колоните на матрицата (не повече от 10000).
- $N \times M$  цели числа – елементите на матрицата, ред по ред
- Неопределен брой (поне една и не-повече от 100) двойки  $(X, Y)$  от цели числа

#### Изход

За всяка двойка от входа трябва да изведете на **стандартния изход** броят на числата от матрицата по-големи или равни на  $X$ , но по-малки или равни на  $Y$ .

Вход	Изход
2 4	5
1 5 10 10	2
2 10 20 99	0
	3
10 99	
2 9	
100 1000	
10 10	

**Пояснение:** По обясними причини данните във входния пример са показани в текстов формат, а не в двоичен, както вашата програма трябва да ги очаква.