

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 18 декември 2011 г.

Задача А. Системата MIU

MIU е формална система, състояща се от низове, в които се срещат само трите букви M, I и U. Тези низове се образуват по следните правила:

A. Ако притежавате низ, чиято последна буква е I, можете да прибавите на края му U.

B. Да допуснем, че имате Mx. Тогава към колекцията си може да добавите Mxx.

C. Ако някой низ от колекцията ви има III, можете направите нов низ с U на мястото на III.

D. Ако в някой низ има UU, можете да го премахнете.

В системата има само една аксиома: Низът MI принадлежи на системата, т.е. вашата колекция съдържа низът MI "по рождение". (От книгата на Хофстатър, „Гюдел, Ешер, Бах – една гирлянда към безкрайността“)

Да се напише програма, която да произвежда низове за колекцията. За да бъде алгоритъмът еднозначен, към правило C допълваме: Заместваме всички тройки III до тогава, докато в низа не останат три последователни букви I. Низът III...I заместваме с U...I, т.е. ако имаме подниз, съдържащ само I, заместваме най-напред най-лявата тройка III. И допълнение към правило D – изтриваме всички двойки UU, докато в низа не останат две последователни букви U.

Входът съдържа много примери, всеки пример на отделен ред. Един пример се състои от редица от главни букви A, B, C и D. За всеки член на редицата се опитваме да приложим съответното правило за текущия низ. Ако низът не отговаря на условието на правилото, не го променяме. Винаги стартираме от низа MI.

За всеки пример на отделен ред отпечатваме низ, получен след прилагане на всички правила от редицата.

Пример:

Вход	Изход
BBACBD	MUIIU
DBBAACBD	MUIIU

Бележка: Получената редица от низове от формалната система е: MI, MII, MIIII, MIIIIU, MUIU, MUIUUUU, MUIIU.

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 18 декември 2011 г.

Задача В. Делене на половина

Дадено е множество S от цели числа. Тегло $t(S)$ на множеството S наричаме сумата от елементите му. Да се раздели множеството на две части S_1 и S_2 така, че теглата на двете части да бъдат максимално близки, т.е. $|t(S_1) - t(S_2)|$ да е най-малкото възможно число.

Всеки пример започва с числото N – броя на елементите на множеството ($2 \leq N \leq 1000$). Следват самите стойности на елементите – цели положителни числа, по-малки от 10000. Входът съдържа много примери.

За всеки пример от входа да се отпечати:

- теглото на по-лекото множество;
- нов ред;
- стойностите на елементите на това множество в нарастващ ред;
- празен ред.

Пример:

Вход	Изход
10	136
3 2 3 2 2 77 89 23 90 11	2 2 2 3 3 11 23 90

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 18 декември 2011 г.

Задача С. Повтарящи се числа

Дадена е редица a_1, a_2, \dots, a_N от цели числа, всяко в интервала от 1 до 100 000, $2 \leq N \leq 10\,000\,000$. Напишете програма, която да намери числото, което се среща най-често в редицата. Ако няколко числа се срещат по равен брой пъти, да се изведе най-малкото от тях.

На първия ред на стандартния **вход** ще бъде зададен броят T на примерите, с които ще бъде тествана програмата. Данните за всеки тестов пример са разположени в два реда. На първия е зададен броят N на числата в редицата, а на втория – N -те числа, разделени с по един интервал.

За всеки тестов пример програмата трябва да **изведе** на отделен ред на стандартния изход най-често срещаното се число.

Пример:

Вход	Изход
2	100
5	1
100 100 200 300 300	
6	
5 4 3 2 1 6	

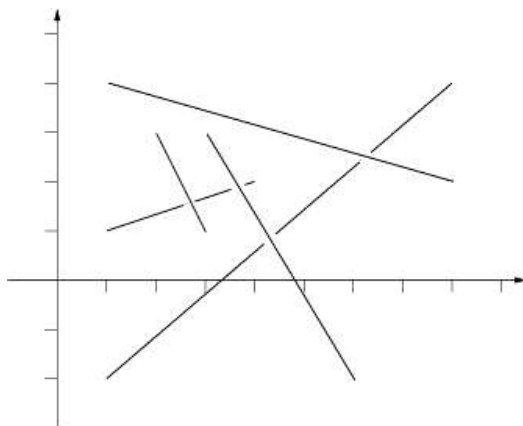
Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 18 декември 2011 г.

Задача D. Гигантско микадо

Тунчо имал n пръчки с различна дължина. Той играе игра, при която ги хвърля една по една произволно по пода. Играта Микадо се състои в изваждането на пръчките една по една без да се мръднат другите. Лесно може да се сетиш, че клечките, които са най-отгоре (върху тях няма други) могат да се извадят най-лесно. Например такава винаги е последната хвърлена. Помогнете на Тунчо да разбере колко им е броя.

Нека приемем, че клечките са с пренебрежимо малка дебелина. **Входът** се състои от няколко теста. Данните за всеки един от тях започват с число $1 \leq n \leq 100\,000$, което показва колко клечки ще има в тази игра. На следващите n реда са записани 4 числа, задаващи координатите на краищата на пръчката ($-1000 \leq x, y \leq 1000$). Пръчките се описват, в реда в който са хвърлени. Както и да ги хвърля Тунчо никога не може да остави повече от 1000 пръчки свободни. Край на тестовете се задава с $n=0$.

За всеки тестов пример, на отделен ред да се **изведе** списък с номерата на пръчките, които са свободни. Този списък трябва да е сортиран по реда на хвърляне и номерата да са разделени със запетая и интервал. Всеки ред завършва с точка.



Пример:

Вход	Изход
5	2, 4, 5.
1 1 4 2	1, 2, 3.
2 3 3 1	
1 -2.0 8 4	
1 4 8 2	
3 3 6 -2.0	
3	
0 0 1 1	
1 0 2 1	
2 0 3 1	
0	

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 18 декември 2011 г.

Задача Е. Единици

Напишете програма, която намира квадрата на число, което се записва с n ($1 \leq n \leq 1\,000\,000$) единици.

От първия ред на стандартния вход се въвежда броят на тестовете. За всеки тест се въвежда стойността на n .

На съответните редове на стандартния изход да се изведат получените резултати като цели положителни числа.

Пример:

Вход	Изход
2	121
2	12345678987654321
9	

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 18 декември 2011 г.

Задача F. Низове

Разглеждаме първите n малки букви от латинската азбука ($1 < n < 20$). Образоваме всички низове с дължина n , съставени от тези букви и такива, че всеки от тези низове не съдържа една и съща буква повече от веднъж. Подреждаме низовете лексикографски (по азбучен ред). Кой е низът, стоящ на k -тото място?

Напишете програма, която въвежда броя на тестовете и за всеки тест въвежда n и k ($0 < k \leq n!$), и извежда резултата на съответния ред в стандартния изход.

Пример:

Вход	Изход
2	bcad
4 9	cab
3 5	

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 18 декември 2011 г.

Задача G. Цифри

Анализираме текст, състоящ се от T параграфа ($1 \leq T \leq 20$), всеки съдържащ N реда ($1 \leq N \leq 100$). Всеки от редовете може да съдържа само цифри и малки букви от английската азбука. За всеки от параграфите искаме да „извлечем“ числата, след премахването на евентуални водещи нули. Ако такива числа съществуват, то след сортирането им в нарастващ ред казваме, че те образуват група.

Всяко число може еднозначно да бъде еднозначно определено, като сканирайки всеки ред от поредния параграф, избираме най-дългата редица, съдържаща само цифри. Например, в реда 01a2b3456cde478 се съдържат числата 1, 2, 3456 и 478.

Входът съдържа T параграфа, всеки започващ с число N , следвано от N -те реда на поредния параграф.

Изходът трябва да съдържа търсените групи в реда, в който сме ги намерили, анализирайки входния текст. Всеки две групи трябва да бъдат разделени с единствен ред, съдържащ само символа '~'.

Пример:

Вход	Изход
3	1
2	3
1o3za4	4
01	~
4	0
43algo0	2
ala002	2
lea2nbu	43
231233	231233
4	~
01bond	0
02james007	1
03bond	2
04contest000	3
	4
	7

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 18 декември 2011 г.

Задача Н. Бира

Разполагаме с S лева ($1 \leq S \leq 1\,000\,000$), които искаме да похарчим за бира. Заведението, в който се намираме, предлага N вида бира ($2 \leq N \leq 20\,000$), всяка на цена L_i ($1 \leq L_i \leq 1\,000\,000$). Ще купуваме комплекти от по 2 вида бира. Кой точно комплекта ще купим, решаваме след като прегледаме всички възможности. Разбира се, само с наличните пари. От колко различни комплекта трябва да направим избора си?

Входът съдържа няколко тестови примера. Първият ред на всеки от тях започва с целите числата N и S . Следват N реда, съдържащи целите числа L_i – цената на i -тия вид бира. Входът завършва с две нули.

За всеки пореден тестов пример трябва да се изведе търсения брой различни комплекти от по 2 вида бира, които бихме могли да си купим с наличните пари, и съответно да изпием след това.

Пример:

Вход	Изход
4 6 3 5 2 1 0 0	4

Забележка: Четирите различни комплекти са: 1 и 3; 1 и 4; 2 и 4; 3 и 4.

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 18 декември 2011 г.

Задача I. Зайци

Редицата от числата на Фибоначи е разгледана от италианския математик Фибоначи през XIII век. Използвал ги, за да моделира нарастването на популацията на зайците. Той забелязал, че броя на двойките зайци, родени през дадена година е равен на сумата от броя на двойките зайци, родени през предходните две години. Т.е. броя на зайците, родени през n -тата година, може да се изрази, чрез следната рекурентна зависимост:

$$F_n = F_{n-1} + F_{n-2}$$



заедно с началните условия $F_0 = 0$, $F_1 = 1$. Първите няколко члена на редицата са:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144 ...

В годините преди прехода един друг математик, чийто интелект и известност по никакъв начин не могат да бъдат сравнени с тези на Фибоначи, се интересувал пък от общия брой двойки зайци в различни периоди $[L, R]$ (с начална и крайна година L и R , включително). Например за периода $[1, 10]$ търсеният брой е 143, за $[3, 33]$ е 9227462, и т.н.

След много часове и безсънни нощи, прекарани в анализиране на резултатите за различни периоди, и сравняването им със статистическите данни, любезно предоставени му от директора на ТКЗС-то в Станке Димитров (днес Дупница), професор Статистичков открил някои шокиращи го разминавания. След дълго умуване и ровене в какви ли не научни трудове, попаднал на брилянтна статия, написана от група съветски учени (Влади Въргала, режисьор на филма „Операция шменти капели“ представи подобна група от учени като „най-великите учени на планетата“). Умът му не го побирал... Гледал и не вярвал на очите си, подобно на познайте кой съвременен професор. Оказало се, че зайците в СССР, както и тези в братска НР България се размножавали по малко по-различен модел (разбира се, нали са от една порода) от този в Италия. Като цяло редицата на Фибоначи се доближавала до този модел, но имало години, в които бройката на двойките зайци се различавала драстично. Такива например били годините след 1986 г. (Чернобилската авария), както и някои други, но да не се отклоняваме твърде много ☺.

По време на прехода професор Статистичков получил предизвикателството на живота си – прилагал математическите си познания, нужни на определени групи хора в тъй наречената

приватизация и изкарвал луди пари от тая работа. И както може да очаквате, дългогодишния му проект, изучаващ популацията на зайците останал на заден план.

Към днешна дата на вас се пада честта да продължете великото дело на вече покойния професор Статистичков, като напишете програма, обработваща множество тестови примери, всеки от тях започващ с броя на годините – N ($1 \leq N \leq 10^4$) и Q ($2 \leq Q \leq 10^6$). Следват Q заявки от следните типове , върху истинската редица на Фибоначи, дефинирана по-горе:

Тип заявка	Очаквано поведение
add I K P	Добавя P заека, P ($1 \leq P \leq 10^4$), зададено в бройна система K ($2 \leq K \leq 36$), към I -тата година, ($1 \leq I \leq N \leq 10^4$). Забележка: Броят на зайците в годините след I -тата година остава непроменен.
get L R S K	Извежда на отделен ред последните S символа ($1 \leq S \leq 20$) на общия брой двойки зайци в периода с начална и крайна години R и L , ($1 \leq L \leq R \leq N$), след преобразуването му в бройна система K ($2 \leq K \leq 36$). Забележка: Ако S е по-голямо от дължината на отговора, тогава трябва да се изведе целия отговор.

Пример:

Вход	Изход
1	1
100 20	143
get 1 1 3 10	143
get 1 10 3 10	43
get 1 10 5 10	9227462
get 1 10 2 10	0011000110
get 3 33 10 10	82ff
get 3 33 10 2	1
get 3 33 4 17	145
add 2 10 2	01
get 1 1 3 10	0210200212
get 1 10 3 10	1136
get 1 10 2 2	5hrye
get 3 33 10 3	2ac6
get 3 33 4 7	9450
get 3 33 10 36	927372692193079001165
get 3 33 4 14	17166978768832
add 10 10 2	
add 14 10 1986	
get 3 33 4 10	
get 1 100 69 10	
get 43 63 14 10	

Нов български университет
Департамент Информатика
Школа „Състезателно програмиране“
СЪСТЕЗАНИЕ, 18 декември 2011 г.

Задача J. Отпуски

След неотдавна приетите промени в закона за отпуските много хора се оказват в следното положение: Трябва да излязат в отпуска, тъй като имат много натрупани дни от предишни години, които не могат да бъдат прехвърлени за следващата година. Ако не ги използват до края на годината, тези дни ще бъдат загубени.

Програмистите в една софтуерна фирма решили вкупом да си вземат по една голяма коледна отпуска... Шефът обаче бил на друго мнение – всеки може да излезе в отпуска, стига това да не удължи крайния срок, в който поредната нова версия на продукта трябва да е готова. Казал на ръководещите проекта да съставят оптимален план, съобразен с оставащите задачи за всеки от програмистите. На базата на този план всеки от тях може да излезе веднага в отпуска, но трябва да се върне на работа в деня, когато трябва да започне първата си задача, според плана.

Програмистът не може да работи по повече от една задача по едно и също време (в практиката това съвсем не е така). За изпълнението на i -та задача са необходими T_i дни и тя трябва да бъде завършена не по-късно от S_i -тия ден, считано от днес (деня с индекс 0!).

Програмата ви трябва да обработи няколко тестови примера. Всеки от тях започва с броя на задачите N ($1 \leq N \leq 1000$) за поредния програмист. Следват N двойки числа – T_i ($1 \leq T_i \leq 1000$) и S_i ($1 \leq S_i \leq 1\,000\,000$), със значение описано по-горе. Входът завършва с числото -1 .

За всеки от примерите, на отделен ред трябва да се изведе търсения отговор.

Пример:

Вход	Изход
4	2
3 5	
8 14	
5 20	
1 16	
-1	

Забележка: В показания пример разполагаме с един програмист, за който има планирани четири задачи, които отнемат по 3, 8, 5 и 1 дена и трябва да бъдат завършени не по-късно от дни 5, 14, 20 и 16, считано от днес.

При оптимален график програмистът трябва да се яви на работа, след два дена и да започне работа по първа задача. След като я приключи, продължава с втората, след това с четвъртата и най-накрая изпълнява третата задача.