

# НАЦИОНАЛЕН ПРОЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Ямбол, 1. юни 2013 г.,  
ГРУПА А, 11 – 12 КЛАС

## ЗАДАЧА А1. САМОТНА ПЕЙКА

Пейката в метрото има  $n$  места, номерирани с числата от 1 до  $n$ . В началото пейката е празна – всичките  $n$  места са свободни. Последователно, един по един, през входа на метрото влизат пътници. Всеки влязъл човек сядва в центъра на първата максимална последователност от свободни места. В случай, че максималната последователност от свободни места съдържа четен брой места, човекът сядва на мястото с по-малък номер измежду двете централни места на последователността.

Например, ако пейката има  $n = 6$  места, първият човек сядва на място с номер 3, защото има единствена максимална последователност от свободни места – от 1 до 6, която съдържа четен брой места, и 3 е мястото с по-малък номер измежду двете централни места (3 и 4) на тази последователност. За втория човек има две последователности от свободни места – от място 1 до място 2 и от място 4 до място 6. Последователността от място 4 до място 6 е максимална, съдържаща 3 места, и вторият човек сядва в центъра ѝ – място 5. Третият човек сядва на място с номер 1 – „централното“ за последователността от свободни места от първо до второ, четвъртият човек – на място 2, петият – на място 4 и шестият – на място 6.

Напишете програма **bench**, която определя мястото, на което сяда определен човек, и кой човек сядва на предварително зададено място.

### Вход

От стандартният вход се въвежда един ред с три цели числа:  $n$   $i$   $p$ , разделени с интервал, където:

- $n$  е общият брой места на пейката;
- $i$  е номер на човек, влязъл в метрото (първият влязъл има номер 1);
- $p$  представлява номер на място от пейката.

### Изход

Програмата трябва да извежда на стандартния изход един ред с две цели числа  $q$  и  $j$ , разделени с интервал, за които е изпълнено, че:

- $i$ -тият човек, влязъл в метрото сядва на място с номер  $q$  и
- $j$ -тият човек, влязъл в метрото сядва на място с номер  $p$ .

### Ограничения

$$1 \leq i, p \leq n \leq 10^{16}$$

В 20% от тестовите е изпълнено  $n \leq 10000$ .

В други 30% от тестовите,  $n = 2^k - 1$ , където  $k > 0$  е цяло число.

#### Пример 1

##### Вход

6 1 5

##### Изход

3 2

#### Пример 2

##### Вход

10 2 6

##### Изход

8 5

# НАЦИОНАЛЕН ПРОЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Ямбол, 1. юни 2013 г.,  
ГРУПА А, 11 – 12 КЛАС

## ЗАДАЧА А2. МОНО

Да наречем „моно“ всяко цяло положително число, в десетичния запис на което се съдържа само една цифра, повторена, евентуално, няколко пъти. Такива са, например: 5, 888, 999999, 22 и т. н. Нека приемем (както обикновено), че „сума“ от едно събираемо е самото то. Тогава всяко естествено число  $n$  може да се представи като сума от монота (тривиално – като сума от  $n$  единици, например). Ние обаче се интересуваме от такова представяне на  $n$  във вид на сума от монота, което съдържа най-малък брой събираеми. Напишете програма **monoa**, която намира едно такова представяне.

### Вход

От стандартния вход се въвежда един ред с цялото положително число  $n$ .

### Изход

Запишете на стандартния изход два реда:

- ред 1: едно цяло положително число  $p$ : намерения най-малък брой събираеми;
- ред 2:  $p$  монота, разделени с интервал, чиято сума е точно  $n$ .

### Ограничения

$n$  има не повече от 11 цифри.

В 20% от тестовите  $n < 200\,000$ .

В 50% от тестовите  $n < 100\,000\,000$ .

### Пример

#### Вход

10203043

#### Изход

7

2 44 777 2222 88888 111111 9999999

### Забележка

Редът на извежданите събираеми няма значение, както и самите събираеми, стига това да са монота със сума  $n$  и да не съществува представяне с по-малко на брой монота.

# НАЦИОНАЛЕН ПРОЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Ямбол, 1. юни 2013 г.,  
ГРУПА А, 11 – 12 КЛАС

## ЗАДАЧА А3. BUBBLES

В дългите и досадни часове по история Ели обича да играе на различни игри на телефона си. В момента нейната страст е играта Bubbles, която има следните правила.

В  $N$  реда, всеки от който съдържа  $M$  колони, са наредени разноцветни балони. Играта се играе в  $K$  хода. На всеки ход Ели може да пукне някой от балоните, като освен него се пукат също така всички съседни по хоризонтала и вертикала балони *със същия цвят*, след това съседните на тях и така нататък, докато не остане съседен непукнат балон от този цвят. След като пукането на всички засегнати балони свърши (което може да считате, че става мигновено), евентуално някои от все още здравите балони ще останат без подпора (тоест балонът под тях е спукан). В тези случаи те почват да падат надолу, докато не стигнат земята или друг, все още непукнат балон.

Нека например имаме следната конфигурация (където различни главни букви отбелязват различни цветове балони) с 5 реда и 20 колони, като сме спукали балона с цвят 'A' (удебелен в показаната по-долу начална конфигурация):

AAAECCCCBBBAAEEPTTR	AAAECCCCBBBAAEEPTTR	AAAE	EETTR		
AAAAECAACAABBAAKAAA	AAAAEC	CAABBAAKAAA	AAAAE	BAAAAKAAA	
ACCCCAESPRIITBBBAAA	ACCC	ESPRIITBBBAAA	ACCC	CCBBBABBBA	
AZCCCAAF <b>AA</b> ABBBASSSAA	AZCC	F	BBBASSSAA	AZCC	ECAAITBASSSAA
ACCCCAAAAAAAAAVAAPAA	ACCC		VAAPAA	ACCC	CCFSPRBBBVAAPAA

Ако играчът е спукал  $P$  на брой балона в текущия ход, той печели  $P \cdot P$  точки - тоест играта насърчава избирането (и създаването!) на по-големи групи балони с еднакъв цвят. Крайният резултат от играта е сумата на точките, получени на всеки от ходовете.

Ели иска да впечатли Станчо, който се слави с това, че не му пука за Ели (каламбурът не е случаен), но за сметка на това пука много добре балоните в тази игра. Знаейки началната конфигурация от балони, както и броя ходове, които може да направи, момичето иска да измисли последователност от пукания, които биха ѝ донесли максимален брой точки. Вие решавате да ѝ помогнете, като напишете програма **bubbles**, която играе тази игра. Защо ѝ помагате, след като най-много вследствие на това тя да се уреди със Станчо, остава мистерия.

### Вход

На първия ред на входния файл **Bubbles.in** ще бъде дадено едно цяло число  $T$  – колко пъти ще играят Станчо срещу Ели. Следват  $T$  теста, отделени с по един празен ред. Всеки тест започва с ред, съдържащ три цели числа  $N_i$ ,  $M_i$ , и  $K_i$  - съответно броя редове и броя колони на таблицата, както и броя ходове, които трябва да направи Ели. Следват  $N_i$  реда, всеки съдържащ по  $M_i$  главни букви от английската азбука ( $\{'A'-'Z'\}$ ). Забележете, че първият ред от балони на входа отговаря на *най-горния* ред! На последния ред ще бъде зададено цялото число  $S_i$ , показващо колко точки е успял да събере Станчо на въпросното ниво. Спрямо тези точки ще бъде оценено и вашето решение (вижте "Оценяване" за пояснение).

### Изход

В тази задача трябва да предадете самия изход, а не решението, с което сте го постигнали.

Предаденият файл трябва да е с име **Bubbles.out** и да съдържа последователност от клетки, които трябва да бъдат натиснати. За всеки тест изведете  $K_i$

# НАЦИОНАЛЕН ПРОЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

Ямбол, 1. юни 2013 г.,  
ГРУПА А, 11 – 12 КЛАС

реда, всеки съдържащ по две цели числа - ред и колона, в които пукате балона на съответния ход. Индексирането започва от нула, като (0, 0) е клетката на таблицата долу вляво.

Забележете, че понякога има смисъл два или повече пъти да цъкате в една и съща клетка, тъй като балоните биха могли да се пренаредят между ходовете. Разрешено е да цъкате в клетки, които вече не съдържат балон, като това не променя нищо и не ви носи точки. Такъв ход е удачен, ако например сте спукали всички балони.

## Ограничения

- ❖  $1 \leq N_i, M_i, K_i \leq 50$
- ❖  $1 \leq S_i \leq N_i * N_i * M_i * M_i$

## Пример

Вход	Изход
2	1 8
	0 0
5 20 4	0 1
AAAECCCCBBBAEEPTTR	3 18
AAAAECAACAABVBAAKAAA	1 0
ACCCCAAESPRITBBVBAAA	1 1
AZCCCAAFAAABVBASSSAA	1 2
ACCCCAAAAAAAAAVAAPAA	1 3
1284	1 4
	1 5
3 23 11	1 6
AAAAAAAAAAAAAAAAAAAAA	1 7
BCBCBCBCBCBCBCBCBCB	1 8
AAAAAAAAAAAAAAAAAAAAA	2 22
1229	0 0

## Обяснение на примера

Първият ход, направен в примерния изход, е примерът от условието, пукащ 18 балона с цвят 'A' в средата на дъската и носи  $18 * 18 = 324$  точки. Вторият ход пука друга група балони с цвят 'A', но този път в лявата страна на дъската. Групата съдържа 10 балона и носи 100 точки. Третият ход пука 14 балона с цвят 'C' и носи 196 точки. Последният ход пука отново 10 балона с цвят 'A', но този път в дясната част на дъската. Сумарно играчът е спечелил  $324 + 100 + 196 + 100 = 720$  точки в играта. Решението му ще получи  $\text{round}(720 / 1284 * 10) = \text{round}(5.60747664) = 6$  точки за този тест.

## Оценяване

За всеки тест, ако вашето решение е събрало  $Y$  точки в играта вие ще получите  $\text{round}(\min(1, Y / S_i) * 10)$  точки за този тест - тоест пропорционално спрямо точките на Станчо, закръглено до най-близкото цяло число в границите  $[0, 10]$ .

## Визуализатор

За ваше удобство е предоставен визуализатор, който тества валидността на изхода ви, като също така показва реалния брой точки, които ще получи предоставеното му като вход решение.

Визуализаторът очаква вход във файл с име **Bubbles.in**, и последователност от ходове във файл с име **Bubbles.out**. И двата файла трябва да са в директорията на изпълнимия файл. В сорса можете да промените дали да се печата дъската след всеки ход, както и колко време да се чака между печатанията.