

XV НАЦИОНАЛНА ОЛИМПИАДА ПО ИНФОРМАТИКА

Общински кръг

25.01.2009 г.

Група В (9.-10. клас)

Задача В1. Последно изключен

По време на газова криза се очаква да се повиши потреблението на електроенергия. Затова Правителството разделило България на N района (номерирани с $1, 2, 3, \dots, N$), така че при претоварване на мрежата, районите да бъдат изключвани един по едни в определен ред: Започва се от първия и след това се изключват през m номера. Например, ако $N = 5$ и $m = 2$, ще се получи следната последователност във времето от изключени райони: $1, 3, 5, 2, 4, 1, 3, 5, 2, 4, 1$ и т.н. Единият от районите бил специален и Правителството искало той да бъде изключен колкото е възможно най-късно. В посочения пример, това е район 4.

Напишете програма **last**, която въвежда броя N на районите ($1 < N < 1000$) и номер на специалния район. Програмата трябва да намери през колко най-малко номера m да се планира извършването на изключванията, така че специалният район да бъде изключен възможно най-късно и все пак да е осигурено, че всеки район ще бъде изключен поне веднъж.

ПРИМЕР 1

Вход	Изход
5 4	2

ПРИМЕР 2

Вход	Изход
20 13	7

Задача В2. Фатален алгоритъм

„Тестове ли? Че то е много просто!” – казва група деветокласници. Защото за тия стандартни тестове (с точно един верен отговор от четири възможни) те успешно прилагат следния „фатален алгоритъм за попълване”:

- намират първото просто число p , не по-малко от предварително определено естествено число M ;
- като започват от отговор a и от числото p , броят последователно отговорите в един въпрос (като ги повтарят, ако трябва: $a, b, c, d, a, b, c, d, \dots$), увеличавайки p с едно на всяка стъпка;
- процесът спира, когато p стане кратно на 13: следващият отговор се избира!
- за начало на броенето при следващия въпрос избират следващото просто число (след текущата стойност на p).

Да разгледаме за пример следния тест:

Въпрос 1: Коя е основната единица за измерване на информация?

a) бит b) квант c) байт d) мегахерц

Въпрос 2: Коя от следващите думи не може да бъде име на променлива в C (Pascal)?

a) zq_1 b) _1zq c) 1_zq d) z_1q

Нека $M=10$. Първото просто число, не по-малко от 10, е 11. Така започва броенето за първи въпрос: 11 – а, 12 – b, 13 – c, стоп!, 13 се дели на 13, избира се отговор **d** (следващият). Сега напред към следващото просто: то е 17. И започва броенето за втори въпрос: 17 – а, 18 – b, 19 – c, 20 – d, 21 – а, 22 – b, 23 – c, 24 – d, 25 – а, 26 – b, стоп!, 26 се дели на 13, избира се отговор **c** (следващият). Така се получава следният набор от отговори: **d c**. И, както виждате, 50% успех!

Лесно, нали? Достатъчно е само да се знае едно число M , за което ще се получи по-голямо съвпадение. Ето тук ще помагате. Напишете програма **fatal**, която намира най-малкото M , за което резултатът от теста ще бъде колкото може по-добър.

Вход:

На първия ред на стандартния вход се въвежда n – общ брой въпроси в теста.

На втория ред се въвеждат правилните отговори на всеки от въпросите: n малки латински букви (всяка **a**, **b**, **c** или **d**), без разделител между тях.

Изход:

Програмата извежда на стандартния изход един ред с намереното число M – най-малкото естествено число, за което „фаталният алгоритъм” дава най-добрия, според вас, резултат.

Ограничения:

$$1 \leq n \leq 50$$

$$M < 1000$$

Всеки въпрос има по четири възможни отговора (**a**, **b**, **c**, **d**), от които само един верен.

ПРИМЕР

Вход	Изход
2	1
ас	

Обяснение на изхода:

Първото просто число, не по-малко от 1, е 2 (1 не е просто!). Тогава броенето протича така: 2-а, 3-б, 4-с, 5-д, 6-а, 7-б, 8-с, 9-д, 10-а, 11-б, 12-с, 13-д, стоп!, отговор **a** се отбелязва за въпрос 1. Следващото просто е 17, така че за втори въпрос броим: 17-а, 18-б, 19-с, 20-д, 21-а, 22-б, 23-с, 24-д, 25-а, 26-б, стоп, даваме отговор **c**. Така постигаме даже 100% успех! 1 е най-малкото естествено число, при което се постига този отличен резултат (то е най-малкото естествено число въобще, 0 не включваме към тях). Същият резултат се постига, разбира се, с начало на броенето $M=2$, (ако щете, и с начало $M=3734$), но ние търсим най-малкото!

Задача В3. КАМЪНИ

Двама приятели играят следната игра. На масата има купчина, съдържаща M камъка в началото. Двамата се редуват да правят ходове, като играчът, който е на ход взема от купчината няколко камъка, спазвайки следните правила:

1. Ако количеството останали на масата камъни се дели на 3, то могат да бъдат взети 1 или 2 камъка;
2. Ако количеството останали на масата камъни при деление на 3 дава остатък 1, то могат да бъдат взети 1 или 3 камъка;

3. Ако количеството останали на масата камъни при деление на 3 дава остатък 2, то могат да бъдат взети 1, 2 или 3 камъка;

Печели играчът, който вземе последния камък.

Двамата приятели играят серия от N игри.

Напишете програма **stones**, която за всеки начален брой камъни в купчината определя кой от играчите (играещ първи или втори ход) може да спечели при правилна игра.

Вход

От първия ред на стандартния вход се въвежда едно цяло положително число N , задаващо броя на игрите в серията;

От втория ред се въвеждат N на брой цели положителни числа, разделени с интервали, задаващи началния брой камъни в купчината за всяка игра от серията.

Изход

На един ред на стандартния изход се извежда символен низ, съдържащ N символа, като в позиция с номер i от този низ се извежда 1, ако в игра с номер i от серията може да спечели играещият първи ход и 2, ако в игра с номер i от серията може да спечели играещият втори ход.

Ограничения

$2 \leq N \leq 100$

$1 \leq M \leq 1000000$

ПРИМЕР

Вход	Изход
3	121
2 3 5	

Обяснение на примера:

В първата игра от серията първият играч печели, тъй като той направо може да вземе двата камъка от купчината.

Във втората игра от серията печеливша стратегия има вторият играч, тъй като първият играч ще вземе 1 или 2 камъка, като и в двата случая вторият играч може да вземе всички останали камъни.

В третата игра от серията печеливша стратегия има първият играч. Той може да вземе 2 и да остави на масата 3 камъка. Тогава вторият играч ще трябва да вземе 1 или 2 камъка, като и в двата случая на следващия ход първият играч може да вземе всички останали камъни.