

Задача Спам (SPAM)

Відомо, що на скриньку електронної пошти часто приходять листи рекламного характеру, листи зі шкідливими вкладеннями або посиланнями, які власник скриньки зовсім не очікував. Такі листи називаються спамом. У Васі є поштова скринька на одному з безкоштовних поштових серверів, і він, як усі нормальні люди, не любить спам. Розмір екрану обмежений і тому перша сторінка списку листів поштової скриньки не може відобразити всю пошту: користувач пошти може бачити не більше M листів і, очевидно, це будуть найсвіжіші з них.

Вася досить довго не перевіряв свою пошту і в його скринці накопичилося N листів, частина з яких є спамом. Якщо він бачить небажану пошту на першій сторінці, то виділяє всі спамерські листи, натискає кнопку «видалити», і виділені об'єкти переміщуються в папку «Видалені». Після цього на першій сторінці відображуються M найсвіжіших листів з решти і серед них може знову опинитися спам. У цьому випадку Вася знову виділяє небажану пошту і видаляє її зі скриньки описаним вище методом. Як тільки на першій сторінці перестане відображатися спам, Вася буде задоволений.

Формат введення/виведення:

Програма *SPAM* спочатку зчитує з клавіатури (стандартного пристрою введення) два цілих числа N і M ($1 \leq N \leq 200000$, $1 \leq M \leq 100000$). Далі зчитується рядок, що містить N чисел (0 або 1), які визначають тип листа (1 - спам, 0 - ні) в порядку від найсвіжіших до найдавніших.

Програма *SPAM* виводить на екран два цілих числа - кількість листів, що будуть видалені, та кількість натискань на кнопку «видалити».

Приклад:

Введення

Виведення

10 4

5 3

0 1 1 0 1 1 0 1 0 1

Задача СумRMQтор (SUMRMQ)

Заповітною мрією винахідника Ніколя Барузова було створення машини (еремк'ютера), основною функцією якої буде генерування послідовності випадкових чисел. Послідовність генерується наступним чином: перше число послідовності дорівнює X_1 , а кожне наступне

обчислюється за формулою $X_i = (X_{i-1} \cdot A + B) \bmod 2^q$, де X_1 , A , B і q - задані цілі числа.

Далі Ніколя задався такою метою: розширити застосування еремк'ютера ввівши виконання ним додаткової функції - пошук мінімального значення у згенерованій послідовності. В процесі тестування виявилось, що еремк'ютер-мінімізатор знаходить мінімум з K послідовно розташованих елементів.

Тепер Ніколя удосконалює свою машину таким чином, що для заданої послідовності з N елементів знаходяться мінімуми для всіх можливих числових відрізків довжиною K (відрізок складають елементи, розміщені підряд), а потім обчислюється їх сума.

Ваша задача написати програму, яка перевіряє правильність роботи машини шляхом підрахунку вказаної суми.

Формат введення/виведення:

Програма *SUMRMQ* зчитує з клавіатури шість цілих чисел, розділених пробілом N, K, X_1, A, B та q ($3 \leq K \leq N \leq 10^7, K \leq 10^6, 0 \leq A, B, X_1 \leq 10^9, 0 \leq q \leq 30$).

Програма повинна вивести одне число - суму мінімумів на всіх відрізках довжини K .

Приклад:

<i>Введення</i>	<i>Виведення</i>
-----------------	------------------

6 2 2 5 9 5	65
--------------------	-----------

Пояснення до тесту:

<i>Згенерована послідовність</i>	2 19 8 17 30 31
----------------------------------	-----------------

<i>Отримана сума</i>	$65 = 2 + 8 + 8 + 17 + 30$
----------------------	----------------------------

Задача Автомагістраль (HIGHWAY)

Пан Бізнесюк збирається у ділову поїздку з рідної Логарифмічної області країни Олімпія у далеке місто Експоненціальськ, причому їхати він буде власним мерседесом. Автомагістраль, що з'єднує Логарифмічну область з Експоненціальськом, складається з N послідовних фрагментів. В середині кожного фрагменту можна їхати або по безплатній дорозі, витрачаючи a_i секунд, або по платній, витрачаючи c_i олімпійських центів та b_i секунд. Між цими фрагментами є транспортні розв'язки, через які можна з'їхати з однієї дороги на іншу. Такі переміщення вимагають q_i секунд (без різниці, з платної дороги на безплатну чи з безплатної на платну). При продовженні руху по тій самій дорозі таких затримок не виникає. Розпочати та завершити рух можна як по платній так і безплатній дорозі, тому розв'язки є лише між першим та другим, другим та третім, ..., $(N-1)$ -м та N -м фрагментами.

Пан Бізнесюк давно знає гасло «час - це гроші» і на даний момент оцінює одну секунду свого часу у K олімпійських центів, а тому пана Бізнесюка цікавить мінімізація величини $P + K \cdot T$, де P - загальна сума сплачених дорожніх зборів, T - загальний витрачений час, зміст величини K пояснений у попередньому реченні.

Формат введення/виведення:

Програма *HIGHWAY* зчитує з клавіатури у першому рядку два цілих числа N ($2 \leq N \leq 60$) та K ($0 \leq K \leq 2012$). Далі у другому рядку зчитується три цілих числа a_1, b_1 та c_1 - час руху по безплатній та платній дорогах першого фрагменту та ціна проїзду по платній дорозі. У кожному з наступних $N-1$ рядків зчитується по чотири цілі числа q_i, a_i, b_i та c_i - час на переміщення між дорогами, час руху по безплатній та платній дорогах цього фрагменту і ціна

проїзду по платній дорозі відповідно. Усі значення a_i , b_i та c_i ($1 \leq i \leq N$) у межах від 1 до 10^{12} .
Усі значення q_i ($2 \leq i \leq N$) у межах від 0 до 10^9 .

Приклад:

Введення

Виведення

5 77

13892

95 17 10000

4 41 17 1000

3 23 17 100

2 17 17 10

1 15 17 1

Пояснення: Для даного прикладу, мінімум величини $P + 77T$ досягається при $P = 1110$, $T = 166$ на маршруті «безплатна (час 95), зміна дороги (час 4), платна (час 17, плата 1000), платна (час 17, плата 100), платна (час 17, плата 10), зміна дороги (час 1), безплатна (час 15)».