

## Задача А. Оохо!

Им'я вхідного файлу: dog.in  
Им'я вихідного файлу: dog.out  
Обмеження по часу: 0.1 секунди  
Обмеження по пам'яті: 64 Мб

Дача Леді охороняється двома собаками, які часто викликають проблеми. Кожного дня до Леді приходять листоноша, молочник та прибиральник сміття. Вони знають, що поведінка собак цілком передбачувана. Коли починається день, одна собака  $A$  хвилин агресивна, а потім  $B$  хвилин спокійна. Аналогічно, інша собака агресивна  $C$  хвилин, а потім спокійна  $D$  хвилин. Обидві собаки повторюють свою поведінку на невизначений термін – агресивний період, потім спокійний період, потім знову агресивний, т.д.

З огляду на час прибуття листоноші, молочника та прибиральника сміття, допоможіть Леді визначити скільки собак (жодна, одна або обидві) нападає на кожного з них. Зверніть увагу, що люди прибувають у  $k$ -ту хвилину, а не рівно через  $k$  хвилин.

### Формат вхідних даних

У першому рядку вхідного файлу знаходиться чотири цілих числа  $A, B, C, D$ . Другий рядок містить цілі числа  $P, M, G$  – хвилини дня, коли листоноша, молочник і прибиральник сміття прибувають до дачі Леді.

Наприклад, якщо  $P = 3$ , то це слід тлумачити як «листоноша прийшов в третю хвилину дня». Усі вхідні дані будуть від 1 до 999.

### Формат вихідних даних

Вихідні дані мають містити три рядка, кожен з яких містить «2», «1» або «0», в залежності від того, скільки собак атакує кожного з наших героїв.

### Приклади

dog.in	dog.out
2 2 3 3 1 3 4	2 1 0
2 3 4 5 4 9 5	1 0 0

### Система оцінювання

В даній задачі кожен тест оцінюється окремо.

## Задача В. Розрахунок

Имя вхідного файла: calculation.in  
Имя вихідного файла: calculation.out  
Обмеження по часу: 0.1 секунди  
Обмеження по пам'яті: 64 Мб

Леді нащедрувала повну кишеню монет по 5, 10, 20 і 50 копійок, і коли вона має сплатити будь-яку суму, то вибирає свої монети таким чином, щоб використати мінімально можливу кількість монет.

Ваше завдання полягає у написанні програми, яка, враховуючи кількість монет по 5, 10, 20 і 50 копійок в кишені Леді, визначає кількість монет кожного номіналу та загальну кількість використаних монет так, що загальна кількість монет, яку вона може використати, була мінімальною. Зауважте, що Леді не завжди зможе заплатити потрібну суму грошей.

### Формат вхідних даних

Перший рядок вхідного файлу містить п'ять цілих чисел — перші чотири це кількості відповідно 5, 10, 20 та 50 копійок, а п'яте число представляє суму, яку Леді має заплатити, в копійках. (Кількість монет кожного типу менша 1 000 000, а сума менша 100 000 000 копійок.)

### Формат вихідних даних

У єдиному рядку вихідного файлу виведіть п'ять чисел: перші чотири — кількість монет кожного типу, які Леді має потратити, а також загальна кількість монет. Якщо розрахуватись Леді не може — виведіть  $-1$ .

### Приклади

calculation.in	calculation.out
2 2 2 2 35	1 1 1 0 3
3 2 0 4 535	-1

### Система оцінювання

Ця задача складається з чотирьох підзадач. Для підзадач виконуються додаткові обмеження, вказані в таблиці. Бали за підзадачу будуть нараховані лише за умови проходження усіх тестів підзадачі.

№ підзадачі	Бали	Обмеження	Примітка
0	0		Тести з умови
1	17	У Леді немає двох різних монет (монет різного номіналу), тобто лише одне з перших чотирьох чисел у вхідному файлі може бути ненульовим.	
2	19	Кількість монет кожного виду не більше 100.	
3	21	У Леді немає монет номіналом в 50 копійок.	
4	43	Обмеження з умови (кількості до $10^6$ , потрібно зібрати до $10^8$ ).	

## Задача С. Рядки в Ужляндії

Им'я вхідного файлу: `strings.in`  
Им'я вихідного файлу: `strings.out`  
Обмеження по часу: 0.5 секунди  
Обмеження по пам'яті: 64 Мб

В Ужляндії знову щось трапилось! Сьогодні відзначає свій день посвяти у лицарі Головний Ужляндський Лицар – Маша. За традицією їй задали Головний Ужляндський Рядок  $S$  довжини  $N$ , що складається з маленьких букв англійського алфавіту. Символи рядка пронумеровані від 1 до  $N$ .

Щоб пройти щорічне випробування, Маші необхідно дістатися останнього елементу рядка, отримавши максимальну кількість балів. Правила випробування та нарахування балів такі:

- Початкова позиція лицаря – елемент рядка з номером 1;
- Лицар може перейти від позиції з номером  $j$  до позиції з номером  $i$  за умови  $1 \leq j < i \leq N$ . За такий перехід він отримає  $(i - j) \cdot |S_i - S_j|$  балів, де  $|S_i - S_j|$  позначає різницю між позиціями цих символів у алфавіті;
- Кінцева позиція – елемент рядка з номером  $N$ .

На жаль, журі випробування не знає яку ж насправді максимальну кількість балів можна отримати. Саме тому воно доручило Вам знайти це значення.

### Формат вхідних даних

У першому рядку вхідного файлу знаходиться одне ціле число  $N$  ( $1 \leq N \leq 10^6$ ) – довжина рядка  $S$ . У другому рядку знаходиться рядок  $S$ , що складається з маленьких букв англійського алфавіту.

### Формат вихідних даних

Виведіть одне число – максимальну кількість балів, яку можна отримати при проходженні випробування.

### Приклади

strings.in	strings.out
4 abca	6
7 abadedc	20

### Система оцінювання

Ця задача складається з п'яти підзадач. Для підзадач виконуються додаткові обмеження, вказані в таблиці. Бали за підзадачу будуть нараховані лише за умови проходження усіх тестів підзадачі, а також всі тести всіх необхідних для неї підзадач. Необхідні підзадачі також вказані в таблиці.

№ підзадачі	Бали	Обмеження	Примітка
0	0		Тести з умови
1	20	$1 \leq N \leq 15$	
2	20	$1 \leq N \leq 1000$ ;	Необхідні підзадачі: 1.
3	10	$1 \leq N \leq 10^6$ ; $S$ може містити лише символи «a» та «b»	Необхідні підзадачі: 0, 1, 2.
4	20	$1 \leq N \leq 10^5$ ;	Необхідні підзадачі: 1, 2.
5	30	$1 \leq N \leq 10^6$ ;	Необхідні підзадачі: 1, 2, 3, 4.

## Задача D. Нова робота для Валл-І

Ім'я вхідного файла:	robot.in
Ім'я вихідного файла:	robot.out
Обмеження по часу:	2 секунди
Обмеження по пам'яті:	64 Мб

Робот Валл-І влаштувався на нову роботу в невеличкий склад, що обслуговує компанію Замазо. Склад вже давно повністю автономний, тож на ньому працює лише 1 робот. На складі є всього  $N$  полиць, які занумеровані цілими числами від 1 до  $N$ . Всі полиці розташовані по одну сторону від доріжки, по якій рухається наш робот. Полиці не обов'язково розташовані в порядку зростання від початку доріжки.

Робот Валл-І має два контейнера: робочий та прокрстинаційний. Спочатку в робочому контейнері Валл-І знаходиться рівно  $N$  коробок, кожен з яких потрібно покласти на одну з полиць. На кожній коробці написаний номер, який означає на яку полицю потрібно її покласти. Кожного дня, коли робот починає працювати і йти повз полиці, він отримує в свій робочий контейнер коробки відсортовані в порядку зростання номерів, що на них написані. Робот може брати для обробки лише ту коробку, яка знаходиться вгорі робочого контейнера.

Робот починає маршрут з полиці, яка розташована на початку доріжки. Зупиняючись біля кожної полиці, робот дивиться на номер полиці. Далі він або проходить мимо, або відкладає певну кількість коробок в прокрстинаційний контейнер і якщо вгорі залишається коробка, яку потрібно поставити на цю полицю, то він її ставить і йде далі. Валл-І не може брати коробки з прокрстинаційного контейнера.

Наприклад, якщо Валл-І підійшов до полиці з номером 5, а верхню коробку потрібно поставити на полицю з номером 2, то він або прибирає в прокрстинаційний контейнер коробки 2, 3 та 4 і викладає коробку 5 на полицю, або проходить до наступної полиці. За один день Валл-І проходить по доріжці  $K$  разів, тому коли він доходить до її кінця, він розвертається і йде в зворотньому напрямку, продовжуючи розставляти коробки.

Програмне забезпечення Валл-І вже застаріле, він не здатен розставити коробки оптимально. Тому ми просимо вас оновити його програмне забезпечення, тобто обрахувати яку максимальну кількість коробок теоретично може розставити Валл-І за 1 робочий день.

### Формат вхідних даних

В першому рядку вхідних даних знаходяться 2 числа  $N$  та  $K$  ( $1 \leq N \leq 10^5$ ,  $1 \leq K \leq 100$ ) – кількість полиць на складі (кількість коробок в робочому контейнері відповідно) і кількість проходів робота за 1 день. В другому рядку вхідних даних знаходиться перестановка з  $N$  цілих чисел – номери полиць в порядку від початку до кінця доріжки.

### Формат вихідних даних

В єдиному рядку вихідних даних має бути єдине ціле число – максимальна кількість коробок, яку зможе розставити Валл-І за 1 робочий день.

### Приклади

robot.in	robot.out
4 1 1 3 2 4	3
6 3 1 5 3 2 4 6	5

### Пояснення до прикладів

**Перший приклад.** За один прохід Валл-І може розставити не більше 3 коробок – покласти коробки 1, 2, 4 або ж 1, 3, 4.

**Другий приклад.** за три проходи по доріжці Валл-І може розкласти 5 коробок. Рухаючись перший раз від початку до кінця він розставити 1, 3, 4, в зворотньому порядку коробку 5, і третій раз, рухаючись зліва, коробку 6.

### Система оцінювання

Ця задача складається з трьох підзадач. Для підзадач виконуються додаткові обмеження, вказані в таблиці. Бали за підзадачу будуть нараховані лише за умови проходження усіх тестів підзадачі.

№ підзадачі	Бали	Обмеження	Примітка
0	0		Тести з умови
1	27	$K = 1$ ; $N \leq 1000$	
2	31	$K \leq 10$ ; $1 \leq N \leq 10\,000$	
3	42	$K \leq 100$ ; $1 \leq N \leq 10\,000$	

## Задача Е. Ліхтарі

Им'я вхідного файлу: lanterns.in  
Им'я вихідного файлу: lanterns.out  
Обмеження по часу: 1.5 секунди  
Обмеження по пам'яті: 64 Мб

Вулицю, на якій живе Леді, освітлюють  $N$  ліхтарів, пронумерованих вздовж вулиці від 1 до  $N$ . Один або кілька ліхтарів, що стоять підряд, назвемо сегментом. Таким чином, загальна кількість сегментів дорівнює  $\frac{N \cdot (N+1)}{2}$ . Сегмент вважається робочим, якщо лампочки в усіх ліхтарях цього сегмента працюють.

З ліхтарями регулярно відбуваються події одного з двох типів:

- в якомусь сегменті через стрибок напруги усі лампочки одночасно перегорають;
- Леді вибирає деякий сегмент і викликає ремонтників, щоб вони замінили на ньому усі перегорілі лампочки.

Після кожної такої події мерія міста, в якому живе Леді, вимагає від неї надати звіт про кількість робочих сегментів. Для покращення показників роботи ремонтників Леді включає у звіт усі сегменти, які є робочими на момент подання звіту або були робочими коли-небудь до цієї події.

Напишіть програму, яка після кожної події визначить кількість сегментів у звіті Леді.

### Формат вхідних даних

У першому рядку вхідного файлу знаходяться два натуральних числа  $N$  і  $Q$  – кількість ліхтарів і кількість подій, що відбулись. Другий рядок вхідного файлу містить  $N$  символів «0» і «1», які описують початковий стан ліхтарів, де «1» означає ліхтар з робочою лампочкою, а «0» – з перегорілою.

У кожному з наступних  $Q$  рядків міститься опис подій у вигляді трьох чисел  $L_i, R_i, C_i$ , які означають, що після цієї події усі лампочки в ліхтарях  $L_i, L_{i+1}, \dots, R_i$

- перегорають при  $C_i = 0$ ,
- замінюють на робочі при  $C_i = 1$ .

В описах усіх подій  $1 \leq L_i \leq R_i \leq N$ , і  $C_i$  приймають значення 0 або 1.

### Формат вихідних даних

У першому рядку вихідного файлу виведіть єдине число – кількість діючих сегментів в початковому стані. Потім по одному в рядку виведіть  $Q$  чисел: для кожної події, що відбулась виведіть кількість сегментів, вказаних у звіті після цієї події.

### Приклад

lanterns.in	lanterns.out
7 4	5
1100101	13
4 6 1	13
3 6 0	19
3 4 1	28
5 7 1	

### Система оцінювання

Ця задача складається з п'яти підзадач. Для підзадач виконуються додаткові обмеження, вказані в таблиці. Бали за підзадачу будуть нараховані лише за умови проходження усіх тестів підзадачі.

№ підзадачі	Бали	Обмеження	Примітка
0	0		Тести з умови
1	9	$1 \leq N \leq 50; 1 \leq Q \leq 150$	
2	11	$1 \leq N \leq 500; 1 \leq Q \leq 250$	
3	15	$1 \leq N \leq 5000; 1 \leq Q \leq 1000$	
4	20	$1 \leq N \leq 50\,000; 1 \leq Q \leq 1000$	
5	45	$1 \leq N \leq 300\,000; 1 \leq Q \leq 300\,000$	