**Ліфтом чи пішки**

|  |  |
| --- | --- |
| **Input file name:** | lift.in |
| **Output file name:** | lift.out |
| **Time limit:** | 100 ms |
| **Memory limit:** | 256 M |

Степан живе на ***N***-му поверсі. Коли Степан заходить в під'їзд, він дивиться, на якому поверсі в цей момент знаходиться ліфт і вирішує, викликати ліфт або піти по сходах.

Сьогодні ліфт знаходиться на ***K***-му поверсі. Степан заходить в під'їзд на 1 поверсі. Він піднімається на один поверх за ***A*** секунд. Ліфт переміщається на один поверх за ***В*** секунд. Часом входу в ліфт і виходу з ліфта, а також переміщення до сходів і назад можна знехтувати.

Допоможіть Степану прийняти рішення, виведіть, за який час він потрапить на свій поверх на ліфті і по сходах, відповідно.

**Вхідні дані:**У єдиному рядку вхідного файлу знаходяться чотири цілих числа ***N, K, A, B (2 ≤ N ≤ 100, 1 ≤ K ≤ 100, 1 ≤ A, B ≤ 103)***.

**Вихідні дані:**Виведіть два цілих числа: час, за який Степан підніметься на свій поверх на ліфті, і час, за який Степан підніметься на свій поверх по сходах.

**Пояснення до прикладу:**У прикладі ліфту необхідна 7 × 3 = 21 секунда, щоб спуститися з 8 поверху і потім 14 × 3 = 42 секунди, щоб піднятися на 15 поверх, де живе Степан. Степану ж необхідно 14 × 5 = 70 секунд, щоб піднятися на 15 поверх сходами.

**Приклад вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **lift.in** | **lift.out** |
| 15 8 5 3 | 63 70 |

**Англійська від Степана**

|  |  |
| --- | --- |
| **Input file name:** | english.in |
| **Output file name:** | english.out |
| **Time limit:** | 100 ms |
| **Memory limit:** | 256 M |

Степан вивчає стародавню англійську мову. Оскільки він робить це в рамках домашнього завдання на завтра, він не встигає детально вивчити його граматику, тому він вирішив для простоти застосувати наступні правила для перекладу сучасних англійських слів у свій варіант стародавньої англійської.

- Усі літери «s», після яких не йде «h» і які не є першими в слові, замінюються на комбінацію «th».

- Якщо перша буква в слові «e», то вона замінюється на «ae».

- Комбінація «oo» замінюється на «ou», причому якщо в слові йде поспіль більше двох букв «o», то з них замінюються тільки перші дві.

Допоможіть Степану перевести кілька слів на свою версію стародавньої англійської мови.

**Вхідні дані:** Перший рядок введення містить число ***N (1 ≤ N ≤ 100)*** - кількість слів, які потрібно перевести. Далі йдуть ***N*** рядків, кожен з яких складається тільки з букв латинського алфавіту. Всі букви кожного слова маленькі, крім, можливо першої, яка може бути заглавною. Довжина кожного слова не перевищує 30.

**Вихідні дані:** Виведіть ***N*** рядків - результат перекладу. Якщо перша літера вихідного слова була заглавною, то такою ж має бути і перша буква перекладеного слова. Інакше всі букви повинні залишитися маленькими.

**Приклад вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **english.in** | **english.out** |
| 3  soon  English  this | soun  Aenglish  thith |

**Подарунки від Діда Мороза**

|  |  |
| --- | --- |
| **Input file name:** | gift.in |
| **Output file name:** | gift.out |
| **Time limit:** | 200 ms |
| **Memory limit:** | 256 M |

Степан справді вірить в Діда Мороза. Він вибирає собі подарунки на новий рік. Степан знає, що Дід Мороз купить йому рівно два подарунки: один нібито від мами, а інший нібито від тата.

У магазині, де Дід Мороз буде купувати подарунки, продається ***N*** подарунків, про кожен подарунок відома його ціна: ціна ***i***-го подарунка дорівнює ***Ai*** грн. Степан знає, що Дід Мороз може витратити на покупку його подарунків не більше ***X*** рублів. Зрозуміло, він хоче отримати якомога більш дорогі подарунки.

Таким чином, він хоче вибрати два різних подарунка з максимальною сумарною ціною, але при цьому вона не повинна перевищувати ***X***.

Допоможіть Степану вибрати собі подарунки.

**Вхідні дані:** Перший рядок введення містить два цілих числа: ***N*** і ***X (2 ≤ N ≤ 100000, 2 ≤ X ≤ 109)***.

Другий рядок введення містить ***N*** цілих чисел: ***A1, A2, . . . , AN (1 ≤ Ai ≤ 109)***. Гарантується, що існує два подарунки з сумарною ціною не більше ***X***.

**Вихідні дані:** Виведіть одне ціле число: максимальну сумарну ціну двох різних подарунків, що не перевищує ***X***.

**Приклад вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **gift.in** | **gift.out** |
| 5 13  2 1 5 4 3 | 9 |

**Гра зi стiльцями**

|  |  |
| --- | --- |
| **Input file name:** | chairs.in |
| **Output file name:** | chairs.out |
| **Time limit:** | 500 ms |
| **Memory limit:** | 256 M |

Вчитель з математики ЛI є органiзатором розважальних конкурсiв в лiцеї. Фiнальним конкурсом буде вiдома всiм "Гра зi стiльцями". Її правила наступнi: є декiлька стiльцiв, їх кiлькiсть повинна бути на один менше, нiж кiлькiсть учасникiв. Наприклад, якщо гравцiв сiм — стiльцiв шiсть. Коли звучить музика, всi бiгають навколо стiльцiв. Як тiльки музика зупиниться, гравцi швидко повиннi сiсти на стiльцi. Кому мiсця не дiсталося той вибуває з гри. Потiм один стiлець забирається i гра продовжується поки не залишиться один учасник.

ЛI знає, що нагорода за фiнальний конкурс дуже цiнна — книжки з математики. Звичайно, вона хоче, щоб їх отримав її найулюбленiший учень, Степан. Як огранiзатор, ЛI може казати коли зупиняти музику, тому вона хоче визначити такий момент, щоб Степан точно встиг зайняти стiлець.

Бiльш формально, поле для гри можна уявляти як неорiєнтований граф з ***N*** вершин та ***M*** ребер, кожне ребро якого має певну довжину. В деяких вершинах графу знаходяться стiльцi, деякi вершини представляють з себе мiсце, в якому може знаходитись якийсь учасник пiд час гри музики, iншi вершини виступають в ролi промiжних, через якi може лежати шлях учасника до стiльця.

ЛI хоче знати найкоротший шлях вiд якоїсь вершини з учасником до якоїсь вершини зi стiльцем. Таким чином, вона зупинить музику тодi, коли Степан буде знаходитись в цiй вершинi i вiн буде мати перевагу в тому, що найкоротший шлях до стiльця в нього буде не бiльший, нiж в будь-якого з iнших учасникiв.

**Вхідні дані:**В першому рядку знаходяться два числа ***N*** i ***M (2 ≤ N ≤ 70000, 1 ≤ M ≤ 200000)*** — кiлькiсть вершин i ребер, вiдповiдно. В другому рядку знаходиться ***N*** цiлих чисел ***Ai (0 ≤ Ai ≤ 2)***. Якщо ***Ai = 0***, то в цiй вершинi нема нi стiльця, нi учасника. Якщо ***Ai = 1***, то в цiй вершинi знаходиться учасник. Якщо ***Ai = 2***, то в цiй вершинi знаходиться стiлець. Гарантується, що хоча б одне з цих чисел дорiвнює одиницi, i хоча б одне дорiвнює двiйцi.

В кожному з наступних ***M*** рядкiв знаходиться по три числа ***Xi, Yi, Zi***, якi позначають, що мiж вершинами ***Xi*** i ***Yi (1 ≤ Xi; Yi ≤ N, Xi ≠ Yi, 0 ≤ Zi ≤ 100000)*** з’єднанi ребром, довжина якого дорiвнює ***Zi*** хвилин. Гарантується, що два ребра з’єднанi не бiльш нiж одним ребром.

**Вихідні дані:**Якщо iснує такий шлях, по якому Степан може добратись з початкового мiсця до стiльця якнайшвидше, то виведiть три числа: ***x, y, d***, якi позначають, що з вершини ***y*** є шлях до вершини ***y*** з сумарним часом ***d*** хвилин. Додатково мають виконуватись умови ***Ax = 1, Ay = 2***.

Якщо такого шляху не iснує виведiть -1.

Якщо iснує декiлька вiдповiдей, виведiть будь-яку з них.

**Приклад вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **chairs.in** | **chairs.out** |
| 6 6  1 1 0 0 2 2  1 3 3  1 2 3  1 5 6  3 6 2  2 3 4  2 4 5 | 1 6 5 |

**Новий алгоритм шифрування**

|  |  |
| --- | --- |
| **Input file name:** | encryption.in |
| **Output file name:** | encryption.out |
| **Time limit:** | 300 ms |
| **Memory limit:** | 128 M |

Фірма, на яку влаштовується Степан, займається розробкою нового наднадійного алгоритму шифрування RSA-SUPER. Степан дізнався, що RSA (абревіатура від прізвищ Rivest, Shamir і Adelman) - криптографічний алгоритм з відкритим ключем. Криптосистема RSA стала першою системою, придатною і для шифрування, і для цифрового підпису. Алгоритм використовується у великому числі криптографічних додатків, включаючи PGP, S / MIME, TLS / SSL, IPSEC / IKE та інших.

Як відомо, в основі алгоритму RSA лежить використання пари простих натуральних чисел ***P*** і ***Q*** і утвореного числа ***N = P \* Q***. Числа ***P*** і ***Q*** називаються ключами шифрування, а число ***N*** - модулем шифрування. Просте число - це натуральне число, яке має рівно два різних натуральних дільники: одиницю і самого себе.

Принциповою відмінністю нового RSA-SUPER алгоритму від RSA алгоритму полягає у виборі ключів. Якщо в реалізації RSA алгоритму потрібнa пара простих чисел ***P*** і ***Q***, то в RSA-SUPER алгоритмі числа ***P*** і ***Q*** повинні бути взаємно простими. Два натуральних числа називаються взаємно простими, якщо вони не мають жодних спільних дільників, крім одиниці.

Для аналізу надійності нового алгоритму власники фірми хочуть дізнатися кількість різних пар ключів ***P*** і ***Q***, таких, що ***1 < P < Q*** і відповідний їм модуль шифрування задовольняє умові: ***N ≤ K***. Саме це нелегке завдання і було дано Степану, а він, звісно просить допомоги у вас.

**Вхідні дані:**Перший рядок вхідного файлу містить одне ціле число ***K (1 ≤ K ≤ 109)***.

**Вихідні дані:**Вихідний файл повинен містити одне ціле число - кількість різних пар ключів ***P*** і ***Q***.

***Оцінювання:***

***K ≤ 300*** - не менше 20 балів

***K ≤ 1000*** - не менше 30 балів

***N ≤ 5000*** - не менше 40 балів

***N ≤ 105*** - не менше 50 балів

***N ≤ 106*** - не менше 60 балів

***Пояснення:*** У першому прикладі: (2,3); (2,5); (3,4)

У другому прикладі: (2,3); (2,5); (2,7); (2,9); (3,4); (3;5)

**Приклад вхідних та вихідних даних:**

|  |  |
| --- | --- |
| **encryption.in** | **encryption.out** |
| 12 | 3 |
| 18 | 6 |