

Задача 1. Праздники

Имя входного файла: holidays.in
Имя выходного файла: holidays.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта
Максимальная оценка: 100 баллов

На планете в звездной системе Альфа Кентавра неделя состоит из A дней, а год — из B дней. Годы нумеруются последовательными натуральными числами: $1, 2, 3, \dots$. Кроме того, годы с номерами C_1, C_2, \dots, C_N являются високосными и состоят из $(B + 1)$ дней. В году дни с номерами D_1, D_2, \dots, D_M являются праздничными. Если праздник попадает на $(B + 1)$ -й день года, то он отмечается только в високосные годы. Первый день первого года является первым днем недели.

Один из жителей планеты решил устроиться на новую работу. В соответствии с заключенным трудовым договором он будет числиться на данной работе в течение E дней, начиная с первого дня 1-го года. По договору он имеет право выбрать один день недели (с 1 по A), который будет для него выходным. Праздничные дни также считаются нерабочими. Житель хочет выбрать себе выходной день таким образом, чтобы за период действия договора у него было максимальное количество нерабочих дней.

Требуется написать программу, которая определяет искомый день недели и вычисляет соответствующее количество нерабочих дней.

Формат входных данных

В первой строке входного файла через пробел записаны числа A и B — количество дней в неделе и в невисокосном году соответственно ($1 \leq A \leq 2500, 1 \leq B \leq 10000$). Во второй строке записано число N — количество високосных лет, и в третьей — номера C_1, C_2, \dots, C_N високосных лет в возрастающем порядке ($0 \leq N \leq 5000, 1 \leq C_1 < C_2 < \dots < C_N \leq 10^7$). В следующей строке число M — количество праздничных дней в году, и на новой строке — D_1, D_2, \dots, D_M в возрастающем порядке ($1 \leq D_1 < D_2 < \dots < D_M \leq B + 1$). В последней строке записано число E ($1 \leq E \leq 10^9$). Известно, что житель заключил контракт не более чем на 10^7 лет.

Формат выходных данных

В выходной файл выведите через пробел два числа — номер дня недели, который выгоднее всего сделать выходным, и соответствующее количество нерабочих дней за период действия договора. Если ответов несколько, то выведите любой из них.

Примеры

holidays.in	holidays.out
7 13 1 2 2 1 14 29	1 8
3 9 0 3 1 4 7 19	2 13

Задача 2. Опечатки

Имя входного файла:	errors.in
Имя выходного файла:	errors.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта
Максимальная оценка:	100 баллов

При наборе текста довольно часто возникают опечатки из-за неправильного «`<<<<` errors.tex нажатия на клавиши. Чаще всего некоторые буквы заменяются на другие, порой возникают лишние, а иногда буквы просто исчезают из слов. В большинстве случаев эти ошибки можно исправить автоматически. Существует метод проверки орфографии, основанный на поиске в словаре слов, *похожих* на проверяемые.

Два слова называются *похожими*, если можно удалить из каждого не более одной буквы так, чтобы слова стали одинаковыми. Например, слова “spot” и “sport” похожи, так как одно и то же слово “spot” можно получить из первого слова без удаления букв, а из второго — удалением буквы “r”. ===== нажатия на клавиши. Например, некоторые буквы оказываются заменены на другие, появляются лишние буквы, некоторые буквы исчезают из слов. В большинстве случаев эти опечатки можно исправить автоматически. В частности, существует метод проверки орфографии, основанный на поиске в словаре слов, *похожих* на проверяемые.

Два слова называются *похожими*, если можно удалить из каждого слова не более одной буквы так, чтобы слова стали одинаковыми, возможно пустыми. Например, слова “spot” и “sport” похожи, так как одно и то же слово “spot” можно получить из первого слова без удаления букв, а из второго — удалением буквы “r”. >>>> 1.14

Требуется написать программу, которая для каждого слова проверяемого текста определяет количество похожих на него слов в словаре.

Формат входных данных

«`<<<<` errors.tex В первой строке входного файла записано натуральное число N — общее количество слов в словаре. В последующих N строках записаны слова, входящие в словарь, по одному на строке. Далее, начиная со следующей строки, записан проверяемый текст, состоящий из слов, разделенных пробелами и/или переводами строк. ===== В первой строке входного файла через пробел записаны натуральные числа $N \geq 1$ — общее количество слов в словаре и $M \geq 1$ — количество слов в проверяемом тексте ($N + M \leq 20\,000$) В последующих N строках записаны слова, входящие в словарь, по одному на строке. Все слова словаря различны. Далее следуют M строк, в которых записаны слова проверяемого текста, по одному слову в строке. >>>> 1.14

Слова состоят из строчных и прописных букв латинского алфавита (прописные и строчные буквы считаются различными). «`<<<<` errors.tex Длина любого слова не превышает 12 букв, общее количество слов в проверяемом тексте и словаре в сумме не превосходит 20 000. Все слова словаря различны. ===== Любое слово состоит не менее чем из одной и не более чем из 12 букв.

>>>> 1.14

Формат выходных данных

«`<<<<` errors.tex Для каждого слова из текста выведите в выходной файл строку, содержащую это слово и количество слов из словаря, на которые оно похоже. Если в словаре имеется единственное похожее слово, то также выведите в этой строке это слово. ===== Для каждого слова из текста выведите в выходной файл строку, содержащую это слово, далее через пробел количество слов из словаря, на которые оно похоже. Если в словаре имеется единственное похожее слово, то также выведите в этой строке это слово (через пробел). >>>> 1.14

Пример

errors.in	errors.out
5 8 father and or mother a Father and mather go o for e walk	Father 1 father and 1 and mather 2 go 1 or o 2 for 1 or e 1 a walk 0

Задача 3. Горнолыжные соревнования

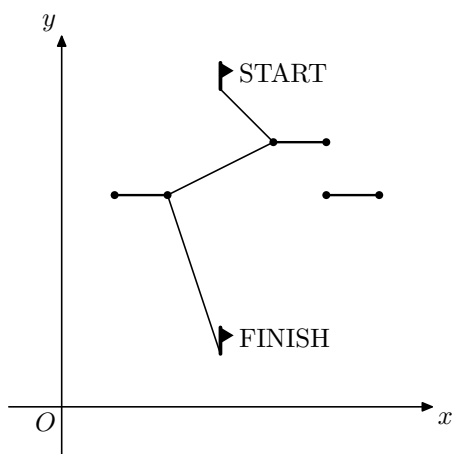
Имя входного файла:	skier.in
Имя выходного файла:	skier.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта
Максимальная оценка:	100 баллов

Горнолыжник, готовясь к соревнованиям, нарисовал на бумаге схему горнолыжной трассы для выбора оптимального маршрута спуска. На схеме расположенные на трассе ворота представлены горизонтальными отрезками. Никакая пара ворот не имеет общих точек.

Маршрут должен представлять собой ломаную, начинающуюся в точке старта на вершине горы и заканчивающуюся в точке финиша у ее подножия. Маршрут выбирается таким образом, что y -координата каждой следующей вершины ломаной оказывается строго меньше y -координаты предыдущей вершины. Один из возможных маршрутов представлен на рисунке.

За каждые ворота, через которые не проходит маршрут, лыжнику начисляются штрафные очки. Общий штраф за спуск по маршруту вычисляется как сумма длины маршрута и штрафных очков за непройденные ворота.

Требуется написать программу, которая определяет, какой минимальный общий штраф горнолыжник может получить при прохождении трассы.



Формат входных данных

В первой строке входного файла задано число N — количество ворот на трассе ($0 \leq N \leq 500$); в следующих двух строках заданы S_x, S_y, F_x, F_y — координаты точек старта и финиша соответственно. В каждой из следующих N строк записаны четыре числа a_i, b_i, y_i, c_i — x -координаты левого и правого концов ворот, y -координата ворот и штраф за непрохождение данных ворот ($F_y < y_i < S_y, c_i$ — целое число, $0 \leq c_i \leq 1000$). Все координаты — целые числа, не превосходящие по модулю 10000.

Формат выходных данных

В выходной файл выведите наименьший возможный общий штраф за прохождение трассы с точностью не менее 4 знаков после десятичной точки.

Пример

skier.in	skier.out
4	7.8126
3 6	
3 1	
5 7 4 1	
4 5 5 10	
1 2 4 5	
2 5 2 0	