

Задача 1. Толя и робот

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Толе на Новый год подарили крутого робота! Его главная особенность заключается в том, что робот понимает простые голосовые команды, такие как «Вверх», «Вниз», «Налево» и «Направо». А еще, робота можно программировать и учить сложным маневрам!

Как и бывает с любой новой игрушкой: ее хочется сразу испытать. Толя придумал несложную задачку для робота: он поместит его внутрь прямоугольного клеточного поля, так, что робот будет находиться внутри него и не на границе поля, а после этого будет давать команды роботу. Робот должен уметь передвигаться по клеткам и отслеживать свое местонахождение в поле. Он должен сообщать – находится ли он строго внутри поля или на его границе. А в случае, если очередная команда Толи выведет игрушку за границу поля, то робот должен сообщить об этом и выключиться.

Осталась одна проблема – Толя плох в программировании на языках высокого уровня, поэтому просит вас помочь ему с написанием алгоритма для робота.

Протокол взаимодействия

Это интерактивная задача, и в ней вам предстоит работать не с файловым вводом-выводом, а со специальной программой – интерактором. Взаимодействие с ней осуществляется через стандартные потоки ввода-вывода.

При старте вашей программе в стандартный поток ввода подаётся четыре целых числа: размеры поля – количество строк и столбцов, соответственно, и координаты начального положения робота ($3 \leq N \leq 10^5$, $3 \leq M \leq 10^5$, $1 < X < N$, $1 < Y < M$). Левая верхняя клетка на плане поля имеет координаты (1, 1).

Ваша программа должна сообщать о том, где находится робот с помощью запросов в стандартный поток вывода.

Запросы могут быть следующими: `inside ?` или `border ?`.

Запрос первого типа мы делаем, если находимся строго внутри поля, второго – на одной из граничных клеток.

Ответом на эти запросы является одна из четырех команд: `U`, `D`, `L` или `R`, что означает соответственно «Вверх», «Вниз», «Налево» и «Направо». Как только робот вынужден будет покинуть поле, он должен вывести сообщение `End of program` и закончить работу.

Гарантируется, что запросов на изменение положение робота не более, чем $2 \cdot 10^5$.

Убедитесь, что вы выводите символ перевода строки и очищаете буфер потока вывода (команда `flush` языка) после каждого выведенного запроса. Иначе решение может получить вердикт `Timeout`.

Примеры

стандартный ввод	стандартный вывод
4 5 2 4 R D L D D	inside ? border ? border ? inside ? border ? End of program
3 3 2 2 U L D D R R U U L D D D	inside ? border ? border ? border ? border ? border ? border ? border ? border ? border ? inside ? border ? End of program

Задача 2. Вадим и столовая

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Вадим больше всего на свете любит плотно поесть. Но, как и любой другой школьник, он ограничен в средствах, и поэтому отказывает себе в удовольствии есть все подряд. Однако, если он очень сильно захочет съесть какое-нибудь блюдо, то не успокоится, пока не получит его.

Сегодня по пути в столовую Вадим составил у себя в голове список блюд, которые он непременно хотел бы купить, и уже прикинул, сколько денег у него уйдет на обед. Каково же было его удивление, когда он узнал, что в его любимой столовой появилась новая акция: комплексные обеды!

Работает эта акция невероятно просто. Каждый комплексный обед задается списком блюд, которые в него входят, а также ценой, которая может оказаться меньше той, которую заплатил бы посетитель столовой, покупая блюда из комплекса по отдельности (но это не гарантируется).

Вадим чувствует, что может сэкономить, но при этом он не хочет отклоняться от намеченного плана. Он намерен взять все блюда, которые есть в его мысленном списке желаний, но заплатить при этом как можно меньше денег. Вадим готов даже взять лишние блюда, если это окажется выгодно. Обратите внимание, что Вадим может купить несколько комплексных обедов. Помогите ему составить заказ.

Формат входных данных

В первой строке входного файла задано число n — количество блюд в мысленном списке желаний Вадима ($1 \leq n \leq 200$).

В следующих n строках задается список желаний. В каждой строке через пробел записаны два числа: a_i и c_i — уникальный номер i -го блюда из списка и его цена, если покупать его отдельно ($1 \leq a_i, c_i \leq 10^9$). Блюда в списке не повторяются.

В следующей $(n + 2)$ -й строке задано число m — количество комплексных обедов в столовой ($1 \leq m \leq 20$).

В следующих $2m$ строках описываются комплексные обеды. Каждый обед задается двумя строками, идущими подряд.

В первой строке задаются два числа k_j и c_j — количество блюд в j -м обеде и цена j -го обеда соответственно ($1 \leq k_j \leq 200$, $1 \leq c_j \leq 10^9$).

Во второй строке записаны k_j чисел: a_{j1}, \dots, a_{jk_j} — номера блюд, входящих в j -й обед ($1 \leq a_{jk_l} \leq 10^9$). Некоторые из блюд могли не встречаться в списке Вадима. Каждое блюдо может встретиться в обеде максимум один раз.

Формат выходных данных

В выходной файл необходимо вывести одно целое число — минимальную сумму, за которую Вадим сможет купить все блюда из своего списка желаний.

Система оценки

В задаче три группы тестов. Баллы за группу начисляются, если пройдены все тесты в группе.

1. $n \leq 10$, $m \leq 10$, все $k_j \leq 10$ (**20 баллов**);
2. $n \leq 20$, $m \leq 20$, все $k_j \leq 20$ (**60 баллов**);
3. $n \leq 200$, $m \leq 20$, все $k_j \leq 200$ (**100 баллов**).

Пример

input.txt	output.txt
6 100 5 200 5 300 5 400 5 500 5 600 3 2 4 6 100 200 300 700 3 5 300 400 500	14

Замечание

В примере из условия Вадиму выгоднее всего взять оба комплексных обеда и купить блюдо с номером 600 отдельно.

Задача 3. Лиза и НОДы

Имя входного файла: `input.txt`
 Имя выходного файла: `output.txt`
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 256 мегабайт

Девочка Лиза еще совсем маленькая, но несмотря на свой возраст уже знает, что такое наибольший общий делитель (НОД). Более того, если вы дадите Лизе два натуральных числа, она без всяких затруднений сможет найти их НОД.

До недавнего времени она этим и занималась: выписывала на листочек произвольные натуральные числа и находила их НОДы. Однако, ей быстро наскучило это занятие, и тогда она придумала новую игру.

Ее правила таковы: Лиза записывает в двух строках $2n$ натуральных чисел, по n чисел в каждой строке. Затем она каким-то образом перемешивает числа в каждой строке, не перенося числа из одной строки в другую, и после этого заполняет третью строку следующим образом: на i -ю позицию третьей строки она записывает НОД двух чисел, которые после перемешивания оказались на i -й позиции в первой и второй строках.

Очевидно, что набор чисел в третьей строке будет зависеть от того, как Лиза перемешивает числа в первых двух строках. Наша юная героиня задалась вопросом: насколько большой может оказаться сумма чисел в последней строке? К сожалению, она еще очень маленькая, чтобы самостоятельно найти ответ на этот вопрос, поэтому она просит вашей помощи.

Формат входных данных

В первой строке входного файла записано натуральное число n ($1 \leq n \leq 200$).

Во второй и третьей строках записаны $2n$ натуральных чисел, по n чисел в каждой строке: $a_1, \dots, a_n, b_1, \dots, b_n$ — это числа, которые Лиза записала на свой листочек в начале игры ($1 \leq a_i, b_i \leq 10^{16}$).

Формат выходных данных

В выходной файл необходимо вывести одно целое число — максимальную сумму, которую можно получить из чисел в третьей строке в конце игры.

Система оценки

В задаче три группы тестов. Баллы за группу начисляются, если пройдены все тесты в группе.

1. $n \leq 8$ (20 баллов);
2. $n \leq 15$ (40 баллов);
3. $n \leq 200$ (100 баллов).

Примеры

input.txt	output.txt
3 2 3 4 6 8 9	9
5 1 3 5 7 9 15 20 30 40 50	13

Замечание

Пояснение ко второму примеру.

Максимальную сумму НОДов мы можем получить, например, следующим перемешиванием:

3	9	5	7	1
15	30	50	20	40
3	3	5	1	1

Задача 4. Кеша и выражения

Имя входного файла: `input.txt`
 Имя выходного файла: `output.txt`
 Ограничение по времени: 1 секунда
 Ограничение по памяти: 256 мегабайт

Недавно на уроке математики Иннокентий узнал про задание приоритета арифметических операций в выражениях. Быстро сделав домашнее задание, он задумался, сколько всего различных выражений длины n существует. Помогите ему!

Определим выражения следующим образом:

1. x — выражение.
2. Если S — выражение, то (S) также выражение;
3. Если S_1 и S_2 выражения, то $S_1 + S_2$ и $S_1 - S_2$ — выражения.

Формат входных данных

Во входном файле задано число N — длина выражения ($0 \leq N \leq 10^6$).

Формат выходных данных

Выходной файл должен содержать одно целое число — количество различных выражений длины N . Так как таких выражений может быть много, то нужно вывести остаток от деления точного ответа на число 998244353.

Система оценки

В задаче три группы тестов. Баллы за группу начисляются, если пройдены все тесты в группе.

1. $N \leq 20$ (**20 баллов**);
2. $N \leq 4000$ (**60 баллов**);
3. $N \leq 10^6$ (**100 баллов**).

Примеры

<code>input.txt</code>	<code>output.txt</code>
3	3
5	11

Замечание

Возможные выражения длины 3:

(x)
 $x + x$
 $x - x$

Возможные выражения длины 5:

$((x))$
 $(x) + x$
 $(x) - x$
 $(x + x)$
 $(x - x)$
 $x + (x)$
 $x + x + x$
 $x + x - x$
 $x - (x)$
 $x - x + x$
 $x - x - x$