

1. Отбор в сборную

ограничение по времени на тест	1 секунда
ограничение по памяти на тест	256 мегабайт

В преддверии участия в Кадетских играх в сборную по спортивному многоборью набирают кадет со всех взводов. От каждого взвода нужно подать по одному участнику. Во взводе все кадеты различного роста. Главное требование отбора: из взвода выбирается кадет среднего роста (т.е. при условии, что во взводе все кадеты различного роста, в шеренге количество кадет ростом ниже данного кадета равно количеству кадет ростом выше него). Трудность заключается в том, что в базе данных сведения о росте кадет содержатся в разных единицах измерения. На выполнение данной задачи командиру взвода отвели слишком мало времени, поэтому он не успеет посчитать вручную. Напишите программу, помогающую командиру взвода определить Фамилию потенциального участника соревнований. Гарантируется, что такой участник всегда имеется и во взводе он всегда один.

Входные данные: в первой строке на вход подается нечетное натуральное число N ($0 < N \leq 10^5$) – число кадет во взводе;

в каждой из n последующих строк вводится фамилия кадета, число a ($m < 300$) и буква (d, v, m, s, f), обозначающая единицу измерения роста (суммарное количество символов во всех фамилиях не должно превышать 10^6)

Выходные данные: в новой строке выводится фамилия кадета, обладающего средним ростом и являющимся кандидатом в сборную.

ПРИМЕЧАНИЕ: для перевода между единицами измерения воспользуйтесь следующими соотношениями:

1 дюйм (d) = 25 мм

1 вершок (v) = 44 мм

1 метр (m) = 1000 мм

1 сантиметр (s) = 10 мм

1 фут (f) = 304 мм

Пример работы программы

Входные данные	Выходные данные
5 Ivanov 2 m Petrov 6 f Sidorov 205 s Pupkin 75 d Mikheev 40 v	Pupkin

2. Переливание крови

ограничение по времени на тест	1 секунда
ограничение по памяти на тест	256 мегабайт

Кадеты изучают тему “Оказание медицинской помощи”. На занятиях преподаватель использует программу-тренажер для расчета банка крови для процедуры переливания. При переливании крови существуют ряд правил:

- лучший вариант переливания крови – переливать кровь той же группы и того же резус-фактора;
- человеку со II и III группой крови можно переливать кровь I группы;

- человеку с IV группой крови можно переливать кровь любой другой группы;
- человеку с I группой можно переливать кровь только I группы;
- человеку с положительным резус-фактором можно переливать кровь отрицательного резус-фактора, но наоборот нельзя.

Кадеты решили написать свой тренажер для тренировки перед контрольной работой. Помогите им написать программу, которая рассчитывает, сколько крови перелить раненому бойцу с определенной группой крови и резус-фактором, если в банке крови есть определённое количество донорской крови разных групп и резус-факторов.

Входные данные: в первых 8-ми строках подаются по 1 числу – количество крови каждой группы и резус-фактора в следующей последовательности: 1-4 группы с положительным резус-фактором, далее 1-4 группы с отрицательным резус-фактором. Далее, с новой строки вводится количество крови, необходимой раненому, ее группа (число от 1 до 4) и резус-фактор (+ или -).

Выходные данные: программа должна вывести состояние банка крови после переливания (если раненому невозможно оказать помощь с данными запасами крови, то банк крови выводится без изменений).

Пример работы программы

Входные данные	Выходные данные
12 13 24 11 7 44 51 9 15 2 +	12 0 24 11 7 42 51 9
12 13 24 11 7 44 51 9 8 1 -	12 13 24 11 7 44 51 9

3. Кто кого переговорит

ограничение по времени на тест	1 секунда
ограничение по памяти на тест	256 мегабайт

На занятиях по обществознанию кадетов очень задела тема занятия и они решили обсудить ее прямо на уроке. Преподаватель, естественно, не в восторге от этой идеи. Для борьбы с «болтунами» у преподавателя есть 2 метода: переговорить их (говорить больше слов в минуту, чем все кадеты в аудитории), либо выгнать с урока за дверь самого болтливого (того, кто произносит и получает от других наибольшее количество слов в минуту). Но возможности болтливых кадет зачастую превышают возможности преподавателя. Напишите программу, определяющую метод борьбы с болтунами.

Входные данные: На вход передается в первой строке натуральное число M – количество слов в минуту ($0 < M < 100$), которые может говорить преподаватель, во второй строке натуральное число N – количество болтливых курсантов ($0 < N \leq 10^5$), в каждой из N последующих строк – фамилии кадетов. С новой строки вводится число L ($0 < L \leq 10^5$) – количество диалогов, далее в последующих L строках вводятся по 3 числа x, y, z – порядковый номер кадета, который говорит, порядковый номер кадета которому он говорит и количество слов, которые он произносит в минуту ($x = y$, то считается, что кадет рассуждает вслух, сам с собой).

Выходные данные: программа должна выдать минимальное количество слов, не превышающее M , которое должен произносить преподаватель в минуту, чтобы переговорить кадетов, либо Фамилию кадета, которого необходимо выгнать.

Пример работы программы

Входные данные	Выходные данные
25 5 Ivanov Petrov Sidorov Pupkin Mikheev 4 1 3 15 2 4 10 1 5 10 3 1 7	Ivanov
40 5 5 Ivanov Petrov Sidorov Pupkin Mikheev 2 1 3 15 2 4 10	26

4. Шифрование

ограничение по времени на тест	1 секунда
ограничение по памяти на тест	256 мегабайт

На экскурсии в музее связи кадеты узнали о различных способах шифрования данных, в том числе криптографических алгоритмах. Кадеты Петров и Пупкин большие любители точных наук, в особенности математики, поэтому их заинтересовал алгоритм RSA, основанный на работе с простыми числами. Вернувшись в расположение, они решили разработать свои открытый и закрытый ключи и обмениваться зашифрованными сообщениями. Для их изготовления необходима пара простых чисел, а точнее произведение простых чисел из промежутка от m до n . Напишите программу, определяющую количество комбинаций, из которых ребята могут выбрать число для своего открытого и закрытого ключа. Под комбинацией понимается произведение двух

различных простых чисел из промежутка от m до n (порядок чисел в произведении не имеет значения).

Входные данные: в одной строке вводится 2 числа m и n - границы числового промежутка из которого выбираются простые числа ($1 \leq m, n \leq 10^5$)

Выходные данные: одно число - количество возможных комбинаций

Пример работы программы

Входные данные	Выходные данные
19 28	1
22 48	21

5. Обмен книгами

ограничение по времени на тест	1 секунда
ограничение по памяти на тест	256 мегабайт

Перед затяжными праздничными выходными в библиотеку учебного заведения завезли новые книги различных жанров. Кадеты (N человек) разобрали все новые книги для чтения. Так как библиотека на выходных не работает, то они передают друг другу прочитанные книги, не возвращая их в библиотеку. В конце каждого дня i -й кадет отдаёт свою книгу p_i -му кадету (в случае, когда $i=p_i$ кадет отдаёт свою книгу самому себе). Гарантируется, что все значения p_i — различные целые числа от 1 до N (то есть p – перестановка). Последовательность p не меняется изо дня в день, она фиксированная.

Например, если $n=6$ и $p=[4,6,1,3,5,2]$, то в конце первого дня книга 1-го кадета будет у 4-го кадета, книга 2-го кадета будет у 6-го кадета и так далее. В конце второго дня книга 1-го кадета будет у 3-го кадета, книга 2-го кадета будет у 2-го кадета и так далее.

Перед вами стоит задача — для каждого i от 1 до N определить число дней, через которое книга i -го кадета вернётся обратно к нему в первый раз.

Предположим, что $p=[5,1,2,4,3]$. Книга 1-го кадета будет передана следующим кадетам:

- после 1-го дня она будет у 5-го кадета,
- после 2-го дня она будет у 3-го кадета,
- после 3-го дня она будет у 2-го кадета,
- после 4-го дня она будет у 1-го кадета.

Таким образом, после четвертого дня книга первого кадета вернётся к своему владельцу. Книга четвёртого кадета вернётся к нему в первый раз сразу после первого дня.

Вы должны ответить на q независимых запросов.

Входные данные: первая строка входных данных содержит одно целое число q ($1 \leq q \leq 200$) — количество запросов. Затем следуют q запросов.

Первая строка запроса содержит одно целое число N ($1 \leq N \leq 200$) — количество детей в запросе. Вторая строка запроса содержит N целых чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$ все p_i различны, т.е. p является перестановкой), где p_i — это ребёнок, который получит книгу от i -го ребёнка.

Выходные данные: для каждого запроса выведите ответ на него: n целых чисел a_1, a_2, \dots, a_n , где a_i — номер дня, в конце которого книга i -го ребёнка вернётся к нему в первый раз в этом запросе.

Пример работы программы

Входные данные	Выходные данные
----------------	-----------------

6	1 1 1 1 1
5	3 3 3
1 2 3 4 5	2 3 3 2 1 3
3	1
2 3 1	2 2 2 2
6	4 4 4 1 4
4 6 2 1 5 3	
1	
1	
4	
3 4 1 2	
5	
5 1 2 4 3	

6. Дежурство по автопарку

ограничение по времени на тест	1 секунда
ограничение по памяти на тест	256 мегабайт

Кадеты в увольнении ходили в кино смотреть детективный фильм. В основе сюжета лежит история проникновения солдат вражеского войска на территорию контрольно-технического пункта (КТП) автопарка военной части и похищение оружия. Кадеты Петров и Пупкин достаточно практичные и считают, что ошибок лучше не допускать, чем их потом исправлять, поэтому они решили найти ответ на вопрос: как возможно было избежать проникновения врага и хищения оружия. Они внимательно изучили обязанности дежурного по автопарку, согласно которым, он следит за сохранностью имущества и военной техники, находящейся под его охраной. Для этого он фиксирует все въезжающие и выезжающие автомобили. Машины, которые выезжали и въехали обратно в автопарк - встают на свои прежние места (если автомобиль въехал впервые, то он встает в конец ряда - место, не принадлежащее никакому из выехавших автомобилей).

В ходе дежурства ведется журнал передвижения транспортного средства, где фиксируются номера въезжающих и выезжающих автомобилей. По истечению дежурства необходимо подать рапорт со сведениями о находящихся в автопарке автомобилях. Кадеты решили, что если автоматизировать этот процесс, то неприятностей, описанных в сюжете, можно избежать. Напишите программу учета въезда/выезда автомобилей. При вводе неверной комбинации Номер автомобиля + команда (при попытке выезда автомобиля, отсутствующего в автопарке либо при въезде автомобиля уже находящегося в автопарке) программа выводит сообщение об ошибке ERROR. Работа автопарка прекращается, начинается сверка автомобилей, перепроверка журнала.

Входные данные: в первой строке вводится натуральное число N – количество автомобилей в автопарке ($0 < N \leq 10^6$);

в каждой из N последующих строк подаются номера автомобилей, находящихся в автопарке; далее в последующих строках вводится код команды и через пробел номер автомобиля, выполняющего данную команду (код команд **in** - въезд, **out** - выезд, **delivery** – сдача наряда), ввод заканчивается командой **delivery**, но количество команд не превышает 10^5 . Номера автомобилей - натуральные числа, не превышающие 10^6 .

Выходные данные: Программа выводит перечень номеров автомобилей, находящихся в автопарке на момент окончания дежурства курсанта.

Пример работы программы

Входные данные	Выходные данные
----------------	-----------------

5 27 56 28 77 48 out 27 out 77 in 44 out 28 delivery	56 48 44
3 27 56 28 out 27 in 44 in 27 delivery	27 56 28 44
3 27 56 28 in 89 out 77 delivery	ERROR

7. Тактика

ограничение по времени на тест	1 секунда
ограничение по памяти на тест	256 мегабайт

На занятиях по тактике кадеты изучают методы перемещения по полю боя. Передвижение на поле боя при действиях в пешем порядке может осуществляться ускоренным шагом или бегом (в полный рост или пригнувшись), перебежками и переползанием. Приемы и способы передвижения на поле боя, их применение зависит от рельефа местности и интенсивности огня противника.

На карте указаны огневые точки противника и расположение прямоугольных заграждений. Количество огневых точек противника всегда 3, количество заграждений различно, оно определяется преподавателем. Места расположения огневых точек и заграждений всегда различно и также определяется преподавателем.

Для определения тактики поведения на поле боя необходимо выяснить, в зоне видимости скольких противников находится боец.

Кадет Петров пообещал преподавателю написать программу для тренажера виртуальной реальности, но его поставили в наряд, а времени осталось мало. Помогите ему написать программу, определяющую количество противников в зоне прямой видимости. Эту программу Петров возьмет за основу для своего тренажера. написал программу для преподавателя, которая позволяет

Входные данные: в первой строке вводится пара целых чисел x, y – координаты бойца ($-10^9 \leq x, y \leq 10^9$);

во второй строке вводятся ещё 3 пары целых чисел $x_1, y_1, x_2, y_2, x_3, y_3$ ($-10^9 \leq x_1, y_1, x_2, y_2, x_3, y_3 \leq 10^9$) – координаты расположения огневых точек противника;
 в третьей строке вводится натуральное число N ($0 < N < 300$) – количество заграждений;
 в n последующих строках вводятся по 2 пары чисел p_1, q_1, p_2, q_2 ($-10^9 \leq p_1, q_1, p_2, q_2 \leq 10^9$) координаты левой нижней и верхней правой вершины заграждения.

Выходные данные: программа должна вывести одно число ($0 \leq k \leq 3$) – количество врагов в прямой видимости.

Пример работы программы

Входные данные	Выходные данные
0 2 3 5 4 -1 5 -4 2 3 2 4 3 -3 3 0 4	3
1 5 1 -2 5 -3 -5 5 2 1 2 3 4 -1 -1 2 0	1

8. Морской бой

ограничение по времени на тест	1 секунда
ограничение по памяти на тест	256 мегабайт

У кадет появилось свободное время, и они решили опробовать новую компьютерную игру «Морской бой». Несмотря на традиционное название, суть игры данной несколько отличается от знакомой всем игры на листах в клетку.

Неподвижная подводная лодка с координатами x_0, y_0 должна поразить движущуюся цель торпедой, имеющей скорость v . Торпеда и цель движутся прямолинейно и равномерно, цель движется без остановок.

Для точного попадания по цели необходимо взять упреждение. Упреждение – это вынос точки встречи снаряда с целью вперед перед целью по ходу ее движения, с учетом скорости движения цели и скорости запускаемого снаряда.

Известны координаты начального положения цели x_1, y_1 , интервал времени движения цели t и координаты цели через заданный промежуток времени – x_2, y_2 .

При расчетах также необходимо учесть, что на подготовку торпеды необходимо затратить время t_1 .

Необходимо определить, в каком направлении нужно выпустить торпеду для поражения цели.

Входные данные: в первой строке подаются координаты подводной лодки x_0, y_0 ; далее, с новой строки вводятся 2 пары чисел (x_1, y_1) и (x_2, y_2) – начальные и конечные координаты цели;

в третьей строке – время движения цели t ;

с новой строки подаются 2 числа – время на подготовку торпеды t_1 и скорость ее движения v .

Координаты всех точек (а также координаты точки встречи торпеды и цели) по модулю не превышают 10^9 , $0 < t, t_1 < 10^9$.

Выходные данные: в новой строке выводится единственное число - азимут направления движения торпеды (в градусах), округленный до целого числа. **Азимут** – угол между северным и направлением на какой-либо удалённый предмет по часовой стрелке.

Пример работы программы

Входные данные	Выходные данные
0 0 2 2 2 1 1 0 2	90
0 0 0 10 1 10 1 0 4	20