

Задача А. АААААА!

На Московской олимпиаде по информатике 10-11 классов в качестве ответа необходимо сдавать не решение задачи на языке программирования, а текстовый файл с ответом на поставленную задачу.

В каждой задаче Московской олимпиады 10-11 классов есть два набора входных данных. В этой задаче входные данные находятся в файлах a1.txt и a2.txt (первый и второй набор соответственно).

Входной файл содержит ровно 1 натуральное число N . В качестве ответа необходимо сдать файл, содержащий N заглавных латинских букв "А".

Ответ на первый тест необходимо сдавать в задачу АААААА!-1, на второй — в задачу АААААА!-2. Обратите внимание, что входные данные являются частью условия задачи и может оказаться так, что вам надо решить задачу только для какого-то очень частного случая.

В этой задаче оба теста проверяются **онлайн**. Во многих других задачах второй (большой) тест во время тура проверяется только на соответствие формату вывода, а баллы выставляются после окончания тура.

Вам дано 2 файла:

a1.txt — первый (маленький) тест к задаче. Оценка за тест: 30 баллов, баллы выставляются только в случае полностью правильного решения. Проверка осуществляется в режиме online (результат виден сразу).

a2.txt — второй (большой) тест к задаче. Оценка за тест: 70 баллов, баллы выставляются только в случае полностью правильного решения. Проверка осуществляется в режиме online (результат виден сразу).

Пример

Входные данные	Результат
3	ААА

Задача В. Факультативы

Ученики школы X любят ходить на факультативы. В начале учебного года им было предложено с помощью электронной анкеты выбрать факультативы, на которые они планируют ходить.

Название каждого факультатива начинается с десятичного числа без ведущих нулей, затем следует точка, пробел и произвольное название факультатива. Например, "22. Programming 101, E.W. Dijkstra" является корректным названием факультатива. Гарантируется, что ни одно название факультатива не содержит в себе подстроки вида «запятая, пробел, десятичное число, точка, пробел».

Названия факультативов, выбранные учеником, записываются в одну строку и между собой разделяются запятой и одним пробелом.

Необходимо посчитать, сколько учеников выбрало каждый из факультативов.

В первой строке входных данных содержится одно натуральное число N — количество учеников в школе. В каждой из следующих N строк перечислены выбранные учеником факультативы.

В первой строке результата должно содержаться K — количество факультативов. После этого должно быть записано K пар строк — название факультатива в первой строке из каждой пары и количество записавшихся на него учеников во второй строке пары. Выводить пары строк с описанием факультатива можно в произвольном порядке.

Вам дано 2 файла:

b1.txt — первый тест к задаче. Оценка за тест: 30 баллов, за каждый неучтенный, лишний факультатив или факультатив с неправильно определенным количеством записавшихся учеников оценка снижается на 10 баллов, однако не может стать меньше нуля. Проверка осуществляется в режиме online (результат виден сразу).

b2.txt — второй тест к задаче. Оценка за тест: 70 баллов, за каждый неучтенный, лишний факультатив или факультатив с неправильно определенным количеством записавшихся учеников оценка снижается на 10 баллов, однако не может стать меньше нуля. Проверка осуществляется в режиме offline (после окончания тура). Ответ принимается на проверку, если количество описаний факультативов совпадает с числом K , записанным в первой строке и каждое описание факультатива состоит из текстовой строки и натурального числа.

Пример

Входные данные	Результат
4	3
22. Prog, 101. Math	1. Algebra, 2 part
1. Algebra, 2 part	1
101. Math	101. Math
22. Prog, 101. Math	3
	22. Prog
	2

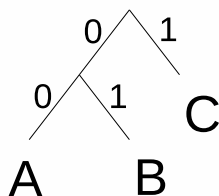
Задача С. Распаковка

Алгоритм Хаффмана — жадный алгоритм оптимального префиксного кодирования алфавита с минимальной избыточностью. При использовании кода Хаффмана каждый символ кодируется целым числом бит, при этом чем чаще встречается символ, тем короче будет его код. Коды Хаффмана обладают свойством префиксности (то есть ни одно кодовое слово не является префиксом другого), что позволяет однозначно их декодировать.

Классический алгоритм Хаффмана на входе получает таблицу частот встречаемости символов в сообщении. Далее на основании этой таблицы строится дерево кодирования Хаффмана (H-дерево), которое является бинарным деревом.

1. Символы входного алфавита образуют список свободных узлов. Каждый лист имеет вес, который равен вероятности появления символа в тексте.
2. Выбираются два свободных узла дерева с наименьшими весами.
3. Создается их родитель с весом, равным их суммарному весу.
4. Родитель добавляется в список свободных узлов, а два его потомка удаляются из этого списка.
5. Одной дуге, выходящей из родителя, ставится в соответствие бит 0 (левой), другой — бит 1 (правой).
6. Шаги, начиная со второго, повторяются до тех пор, пока в списке свободных узлов не останется только один свободный узел. Он и будет считаться корнем дерева.

Например, если в сообщении встречалась буква А с частотой 0.3, буква В с частотой 0.3 и буква С с частотой 0.5, то одно из H-деревьев может выглядеть как на рисунке.



Символ А будет кодироваться как 00, В — 01, С — 1. Сообщение АСВААВВССС будет закодировано как 0010100000101111. Каждый символ имеет уникальный код.

Это дерево можно записать в Newick-подобном формате как ((A,B),C). Такое описание однозначно задает дерево.

В форме Бэкуса-Наура формат описания дерева выглядит так:

`<лист> ::= <символ>`

`<дерево> ::= <символ> | (<дерево>, <дерево>)`

По заданному описанию дерева и закодированной последовательности вам необходимо восстановить исходную последовательность. В дереве листьям могут соответствовать символы с кодами от 32 до 126 включительно.

В первой строке входных данных задается число N — количество последовательностей, которое нужно распаковать. Следующие $2 \times N$ строк содержат описание одного набора упакованных данных: описание дерева и зашифрованного с помощью него сообщение.

Для каждого из описаний необходимо вывести распакованную последовательность символов. Если вы не можете распаковать какое-либо описание — запишите в соответствующей строке "NO ANSWER".

Вам дано 2 файла:

c1.txt — первый тест к задаче. Оценка за тест: 30 баллов, за каждую неправильную распакованную последовательность оценка снижается на 10 баллов, однако не может стать меньше нуля. Проверка осуществляется в режиме online (результат виден сразу).

c2.txt — второй тест к задаче. Оценка за тест: 70 баллов, за каждую неправильную распакованную последовательность оценка снижается на 1 балл, однако не может стать меньше нуля. Проверка осуществляется в режиме offline (после окончания тура). Ответ принимается на проверку, если количество строк в ответе совпадает с количеством описаний во входном файле.

Пример

Входные данные	Результат
2 (a,b) 001 ((A,B),C) 0010100000101111	aab ACBAABVCCC

Задача D. Интеграл

Жители острова Мумба-Юмба все как один участвуют в строительстве первой в истории острова ракеты под названием «Интеграл».

Чтобы ракета долетела до цели, необходимо тщательно отслеживать изменение показаний датчиков. Датчики на «Интеграле» могут возвращать значение 0 или 1. Чтобы отследить изменение показаний необходимо разработать схему, которая на единственном выходе будет изменять свое значение на противоположное при изменении показаний одного из датчиков (любого). Например, если показания датчиков были (0, 1, 0) и схема на выходе возвращала 0, то при показаниях датчиков (1, 1, 0), (0, 0, 0) или (0, 1, 1) она обязательно должна на выходе возвращать 1.

На острове действует программа импортозамещения, поэтому систему отслеживания изменений решено производить из собственных компонентов. Электронная промышленность Мумба-Юмбы не очень развита и может изготовить только простейшие логические элементы: «и», «или» и «не».

Вам необходимо построить две схемы, на вход которым поступают показания датчиков, а выходы должны соответствовать описанным выше условиям.

Всего на «Интеграле» установлено N датчиков.

В первой строке выведите общее число элементов схемы K (не более 5000), включая N входов и один элемент для вывода результата. Входы в схеме имеют номера от 1 до N .

Каждая из следующих $K - N$ строк описывает логический элемент (они получают номера начиная с $N + 1$ подряд в порядке описания) и начинается с одного из символов "&", "|", "!", "#", обозначающих, соответственно «и», «или», «не» и выход схемы. После каждого из символов следует одно или два числа — номера элементов схемы (входов или описанных в других строках логических элементов), результат которых принимает на вход этот элемент. После «или» и «и» указываются два числа, после «не» и выходного элемента — одно. Выходной элемент в схеме должен быть ровно один.

Выход элемента может быть входом для нескольких других элементов.

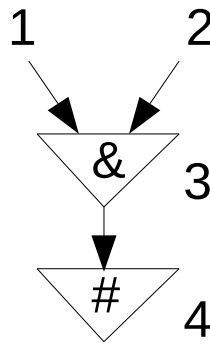
В первом (маленьком) тесте необходимо построить схему для 5 датчиков. Оценка за тест: 30 баллов, баллы выставляются только в случае, если схема выдает правильный результат во всех случаях. Проверка осуществляется в режиме on-line (результат виден сразу).

Во втором (большом) тесте необходимо построить схему для 15 датчиков. Максимальная оценка за тест: 70 баллов, оценка вычисляется по формуле $Score = 70 \times \frac{MinCount}{Count}$, где $MinCount$ — минимальное число использованных элементов среди всех участников и жюри, а $Count$ — количество элементов в решении участника. Входы также считаются за логические элементы. Баллы выставляются только в случае, если схема выдает правильный результат во всех случаях. Проверка осуществляется в режиме offline (после окончания тура). Во время тура проверяется, что в схеме не более 5000 элементов, ровно один выход, ни один из элементов не принимает ничего из выхода, а также соблюден формат описания операций.

Пример

Входные данные	Результат
2	4 & 1 2 # 3

Ответу соответствует схема:



В этой схеме всего 4 элемента: 2 входа (они имеют номера 1 и 2), один элемент «и» (имеет номер 3, на вход получает выходы элементов 1 и 2) и выходной элемент 4, который на вход получает выход элемента 3.

Обратите внимание, что эта схема решает задачу неправильно. Например, при переходе из состояния $(0, 0)$ в состояние $(0, 1)$ значение функции не изменится.

Задача Е. Распределение по группам

При распределении студентов по группам руководство образовательной программы X хочет добиться того, чтобы группы получились как можно более равными по силе. Для этого перед началом учебного года проводятся два тестирования: по математике и программированию, по результатам которых и происходит распределение.

Всего на курсе планируется K групп, по которым нужно распределить N студентов, причём количество людей во всех группах должно быть равно N/K . Гарантируется, что N делится на K нацело. Сила группы характеризуется двумя числами: суммарным количеством баллов на тестировании по математике и тестировании по программированию у студентов этой группы.

После распределения можно вычислить лучшую и худшую группы по математике и программированию. Среди всех групп найдём группы, где количество баллов по математике и программированию максимально и минимально. Максимальную сумму баллов по программированию среди всех групп обозначим P_{max} , минимальную — P_{min} . Аналогично для математики определим M_{max} и M_{min} . Качество распределения обозначим за $Q = P_{max} - P_{min} + M_{max} - M_{min}$. Чем меньше величина Q , тем более равномерно распределены группы.

В первой строке входных данных содержится два числа N и K — количество студентов и групп соответственно. В следующих N строках содержатся результаты тестирований студентов, в i -й строке содержатся два целых числа M_i и P_i — результаты тестирований по математике и программированию i -го студента соответственно.

Выведите распределение студентов по группам: N строк, i -я из которых содержит число от 1 до K — номер группы, в которую распределен студент номер i . Студенты нумеруются в том порядке, в котором они перечислены во входном файле.

Решение с наименьшим среди участников и жюри Q_{min} будет получать 30 и 70 баллов (за первый и второй тест соответственно), остальные решения будут получать $\max(30(70) - (Q - Q_{min}), 0)$ баллов, где Q — качество распределения участника.

e1.txt — первый тест к задаче. Оценка за тест: 30 баллов. Баллы выставяются только в случае, если каждый студент распределен в какую-то группу и размеры всех групп равны, количество баллов вычисляется по формуле, приведенной выше. Проверка осуществляется в режиме online (результат виден сразу).

e2.txt — второй тест к задаче. Оценка за тест: 70 баллов. Баллы выставяются только в случае, если каждый студент распределен в какую-то группу и размеры всех групп равны, количество баллов вычисляется по формуле, приведенной выше. Проверка осуществляется в режиме offline (после окончания тура). Ответ принимается на проверку если каждый студент распределен в какую-то группу и размеры всех групп равны.

Пример

Входные данные	Результат
4 2	2
1 4	1
2 3	1
3 2	2
4 1	

В примере, если распределить первого и четвёртого студентов в первую группу, а второго и третьего — во вторую, то в каждой группе сумма баллов по каждому тестированию равна 5, итоговое Q равно 0.

Задача F. Путешествие Кота

В верхней левой клетке квадратного поля $N \times N$ сидит Кот. Известный видеоблоггер Полиграф Шариков хочет снять M разных видео с путешествием Кота в правый нижний угол поля, раскладывая в некоторых из клеток поля кильку.

Кот очень ленив — он готов пройти только по кратчайшему пути (содержащему наименьшее количество клеток). При этом он готов переходить в соседнюю клетку только если в ней лежит килька. В начальную клетку также нужно обязательно положить кильку.

Чтобы видео получилось интересным необходимо разложить кильку таким образом, чтобы у Кота было ровно K способов пройти свой маршрут. Помогите Полиграфу придумать, в какие клетки необходимо положить кильку, чтобы выполнялись все приведенные выше условия.

В первой строке входных данных записано 2 числа: N и M . В следующей строке записано M чисел K_i — требуемое количество способов.

Выведите M таблиц размером $N \times N$, где 0 обозначает отсутствие кильки в клетке, а 1 — ее наличие. В i -ой таблице должно существовать ровно K_i различных путей. Если вы не можете решить задачу для какого-то из случаев, то можно вывести таблицу, состоящую из нулей.

Обратите внимание, что вам нужно решить задачу для конкретных N и K , а не в общем случае.

Вам дано 2 файла:

f1.txt — первый тест к задаче. Оценка за тест: 30 баллов. Проверка осуществляется в режиме online (результат виден сразу). Каждая верно сгенерированная таблица оценивается в 6 баллов.

f2.txt — второй тест к задаче. Оценка за тест: 70 баллов. Проверка осуществляется в режиме offline (после окончания тура). Ответ принимается на проверку, если выведено ровно $N \times N \times M$ чисел и все они равны 0 или 1. Каждая верно сгенерированная таблица оценивается в 14 баллов.

Пример

Входные данные	Результат
3 2	1 1 0
4 2	1 1 1
	0 1 1
	1 1 0
	1 1 0
	0 1 1

Задача G. Генерация пил

В этой задаче вам необходимо посчитать количество пилообразных последовательностей длины N , состоящих из чисел от 1 до N (включительно).

Под пилообразной последовательностью будем понимать такую последовательность, где первое число меньше второго, второе — строго больше своих соседей (первого и третьего), третье — строго меньше своих соседей и т.д. Числа, стоящие на нечетных местах (при нумерации с единицы) должны быть меньше своих соседей, а стоящие на четных местах — больше. У крайних чисел есть только один сосед.

В единственной строке входных данных содержится одно натуральное число N .

В единственной строке результата должно содержаться K — количество пилообразных последовательностей по модулю 10^7 .

Вам дано 2 файла:

g1.txt — первый тест к задаче. Оценка за тест: 30 баллов. Проверка осуществляется в режиме online (результат виден сразу).

g2.txt — второй тест к задаче. Оценка за тест: 70 баллов. Проверка осуществляется в режиме offline (после окончания тура). Ответ принимается на проверку, если в ответе содержится ровно одно целое число.

Пример

Входные данные	Результат
3	5

Пилообразные последовательности длины 3: (1, 2, 1), (1, 3, 1), (1, 3, 2), (2, 3, 1), (2, 3, 2).