

Задача A. На самокате по лужам

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

Катя катается на электросамокате во дворе. Как удачно, что ее двор расчерчен в прямоугольную сетку, а Катя ездит по ее узлам. Преодолеть расстояние между двумя соседними узлами Катя может за 10 секунд.

Также во дворе Кати есть лужи. Лужи являются прямоугольниками, углы которых совпадают с узлами сеток. Внутри лужи скорость передвижения отличается от стандартных 10 секунд на ребро (включая границу).

Вам даны данные о n лужах, от вас требуется найти для Кати самый быстрый путь из A в B .

Формат входных данных

В первой строке находятся координаты стартовой и конечной точки маршрута x_s, y_s, x_f, y_f .

Во второй строке вводится одно натуральное число n ($1 \leq n \leq 1000$) — количество луж в Катином дворе.

В следующих n строках вводится пятерка чисел x_l, y_b, x_r, y_t, t_i ($10 \leq t_i \leq 10^8$) — координаты левого нижнего, правого верхнего угла лужи, а также время, за которое преодолевается одно ребро сетки в луже.

Лужи не пересекаются и не касаются. Старт и финиш находятся в разных узлах и не находятся в лужах или на их границах. Все координаты целые, неотрицательные и не превосходят 10^8 .

Формат выходных данных

Выведите одно число — оптимальное время, за которое можно добраться от старта до финиша.

Пример

стандартный ввод	стандартный вывод
1 6 15 3 4 2 1 3 7 44 5 2 10 4 33 8 5 11 9 22 12 1 14 8 11	192

Задача В. Мул

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

У вашего мула есть n мешков для перевозки. При перевозке мешка с номером i вы получите a_i монет (правда, a_i могут быть и отрицательными).

Изначально вы собираетесь повесить на своего мула все мешки (разумеется, ваш мул их выдерживает). Ваш мул умеет сбрасывать с себя мешки определенным образом. А именно, если он сбрасывает с себя мешок k , то он также сбрасывает все мешки на позициях $2 \cdot k, 3 \cdot k, \dots$. Если какой-то мешок был сброшен, мул его больше не собирает перевозить.

Мул хочет сбросить такие мешки, чтобы итоговый заработок был максимален. Помогите ему.

Формат входных данных

В первой строке вводится целое положительное число n ($1 \leq n \leq 100$) — количество мешков у вашего мула. В следующей строке вводится n целых чисел a_i ($-10^9 \leq a_i \leq 10^9$) — прибыль при перевозке i -о мешка.

Формат выходных данных

Выведите одно число — максимальный возможный заработок. Обратите внимание, что он может оказаться отрицательным.

Примеры

стандартный ввод	стандартный вывод
6 1 2 -6 4 5 3	12
6 100 -100 -100 -100 100 -100	200
5 -1 -2 -3 -4 -5	0
2 -1000 100000	99000

Задача C. СНМ

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Ваша задача — реализовать **Persistent Disjoint-Set-Union**. Что это значит?

Про **Disjoint-Set-Union**:

Изначально у вас есть n элементов. Нужно научиться отвечать на 2 типа запросов.

- $+ a b$ — объединить множества, в которых лежат вершины a и b
- $? a b$ — сказать, лежат ли вершины a и b сейчас в одном множестве

Про **Persistent**:

Теперь у нас будет несколько копий (версий) структуры данных **Disjoint-Set-Union**.

Запросы будут выглядеть так:

- $+ i a b$ — запрос к i -й структуре, объединить множества, в которых лежат вершины a и b . При этом i -я структура остается не измененной, создается новая версия, ей присваивается новый номер (какой? читайте дальше)
- $? i a b$ — запрос к i -й структуре, сказать, лежат ли вершины a и b сейчас в одном множестве

Формат входных данных

На первой строке 2 числа N ($1 \leq N \leq 10^5$) и K ($0 \leq K \leq 10^5$) — число элементов и число запросов. Изначально все элементы находятся в различных множествах. Эта начальная копия (версия) структуры имеет номер 0.

Далее следуют K строк, на каждой описание очередного запроса. Формат запросов описан выше. Запросы нумеруются целыми числами от 1 до K .

Пусть j -й из K запросов имеет вид « $+ i a b$ ». Тогда новая версия получит номер j . Запросы вида « $? i a b$ » не порождают новых структур.

Формат выходных данных

Для каждого запроса вида $? i a b$ на отдельной строке нужно вывести YES или NO.

Пример

стандартный ввод	стандартный вывод
4 7	NO
+ 0 1 2	YES
? 0 1 2	YES
? 1 1 2	YES
+ 1 2 3	NO
? 4 3 1	
? 0 4 4	
? 4 1 4	

Задача D. Зачет с блэкджеком

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Это интерактивная задача.

Мы решили сыграть с вами в Блэкджек. Расскажем правила:

- Вы играете на колоде из 52 карт, 6 копий каждой карты. У каждой карты есть значение. Карты с числами имеют соответствующее значение, король, дама, валет имеют значение 10. Значение туза равно одновременно 11 и 1 (в зависимости от вашей руки — если сумма в руке больше 21, то 1, иначе 11).
- Дилер сдает вам две карты, а себе одну карту.
- После этого вы можете набирать сколько угодно карт, пока ваша сумма не больше 21. Остановиться можно в любой момент.
- Кроме добора карты вы можете удваивать свою ставку перед добором.
- После этого дилер набирает руку себе. Дилер пользуется правилом «Добирать, пока в руке сумма меньше 17»
- В этот момент раунд завершается и идет подсчет выигрышей. Если в вашей руке 22 и больше, то считается, что ваша сумма равна нулю. Побеждает тот, кто набрал сумму ближе к 21. Если ваша сумма совпала с суммой дилера, ставка возвращается вам в полном объеме. Если вы обыграли дилера, то вам вам вы получаете выигрыш 1:1.

У вас есть одна московская премия победителя всероса (500.000 рублей). Вам нужно заработать один биткоин (4.200.000 рублей) за не более, чем 10000 раундов.

Протокол взаимодействия

В начале каждого хода интерактор пишет «bet». После этого ваша программа должна вывести натуральное число рублей, которое вы ставите. Разумеется, нельзя поставить больше, чем у вас есть. Выведите -1, если хотите завершить игру с текущим балансом.

После этого дилер итеративно выводит вам стол (цифры от 2 до 9 означают значение, «A», «J», «Q», «K», «0» означают туз, валет, даму, короля, десять соответственно) и спрашивает, хотите ли вы взять еще одну карту. Если вы хотите взять еще одну карту, то выведите «hit». Если хотите удвоить ставку и взять карту, выведите «double». Иначе выведите «break». Нельзя просить добрать карту, если ваша текущая сумма больше 21.

Следующим этапом дилер добирает карты себе (дилер играет честно по правилу «добирать, если текущая сумма не больше 17») и выводит вам свой окончательный стол. После этого карты уходят в сброс и начинается следующий круг.

Иногда перед «bet» будет выводиться «shuffle». Это значит, что дилер взял новую колоду из 52 карт и перемешал ее.

Поскольку это интерактивная задача в казино, то любое нарушение протокола будет караться случайным вердиктом. А именно, если ваш вердикт не «OK», то это будет случайный вердикт из «WA», «PE».

Пример

стандартный ввод	стандартный вывод
bet	500000
A#K2	hit
A#K28	break
A0#K28	-1
bet	

Задача E. ТСП

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Это неточная задача с одним открытым тестом.

Вам дан граф городов США (смотри ссылки в примечании). Вам нужно составить такой маршрут для вашего квадрокоптера, чтобы он пролетел этот маршрут по прямой и побывал во всех городах как минимум по одному разу.

А именно, от вас требуется вывести такую последовательность городов, что длина маршрута будет минимальна. Учтите, что все неровности планеты уже учтены и вам дан граф действительно кратчайших расстояний.

Формат входных данных

https://people.sc.fsu.edu/~jburkardt/datasets/cities/usca312_dist.txt — Матрица попарных расстояний

https://people.sc.fsu.edu/~jburkardt/datasets/cities/usca312_xy.png — Визуализированный датасет

Формат выходных данных

Выведите $n = 312$ чисел — последовательность городов для посещения.

Дальше выведите матрицу $n \times n$ — матрицу попарных расстояний графа.

Замечание

У преподавателей на текущий момент еще нет решения этой задачи. Ожидается, что модельное решение будет давать ответ, в худшем случае в полтора раза длиннее, чем оптимальный. Вам разрешается проиграть модельному решению не более, чем в полтора раза (*возможно, мы расширим квоту*). Модельное решение будет либо решением преподавателя, либо лучшим из решений участников.

Сдавайте в систему код, который использовали для получения ответа. Не обязательно его запускать, достаточно закомментировать. Если ваше решение просто выводит прекальканную перестановку или делает другую неосмысленную работу, это решение имеет все шансы быть оцененным в ноль баллов.