



*Эффективная и слаженная
командная работа — залог успеха!*

ХІХ РЕСПУБЛИКАНСКАЯ КОМАНДНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ ПО ПРОГРАММИРОВАНИЮ

Якутск
4–5 марта 2022 г.

XIX Республиканская командная олимпиада школьников по программированию. — Якутск, 2022.

Сборник содержит условия задач XIX Республиканской командной олимпиады школьников по программированию и возможные варианты решений. Олимпиада проводилась дистанционно 4–5 марта 2022 г.

СПОНСОРЫ ОЛИМПИАДЫ



Генеральный директор
ГК «КопирТехСервис»
НИКИФОРОВ Александр Тарасович



СЕВЕРО-ВОСТОЧНЫЙ ФЕДЕРАЛЬНЫЙ
УНИВЕРСИТЕТ ИМЕНИ М.К. АММОСОВА

Преподаватели СВФУ
имени М.К.Аммосова
НИКОЛАЕВА Наталья Васильевна,
ЛЕВЕРЬЕВ Владимир Семенович



Руководитель МАУ «Кинотеатр
Лена» ГО «город Якутск»
ЕГОРОВ Алексей Алексеевич

ОРГКОМИТЕТ ОЛИМПИАДЫ

ФЕДОРОВ Михаил Прокопьевич

*заместитель ректора Северо-Восточного федерального
университета имени М. К. Аммосова по кадровой политике —
председатель*

ПАВЛОВ Василий Климович

ректор Малой академии наук РС (Я) — заместитель председателя

ПЕТРОВА Дария Александровна

*и.о. ведущего специалиста департамента по гос. политике
общего образования, воспитания и дополнительного образования
Министерства образования и науки РС(Я)*

НИКОЛАЕВА Наталья Васильевна

*доцент ИМИ СВФУ, заведующая кафедрой информатики
Малой академии наук РС (Я)*

СЕМЕНОВА Галина Александровна

*проректор по учебно-организационной и воспитательной
работе Малой академии наук РС (Я)*

ОБУДОВ Владислав Антонович

диспетчер Малой академии наук РС (Я)

ЖЮРИ ОЛИМПИАДЫ

ПАВЛОВ Никифор Никитич

*к. ф.-м. н., доцент кафедры «Информационные технологии»
ИМИ СВФУ — председатель*

БУЛАТОВ Василий Алквиевич

разработчик, ООО «Яндекс.Такси Технологии» (Яндекс Такси)

ГАВРИЛЬЕВ Валерий Валерьевич

разработчик, ООО «Яндекс.Технологии» (Яндекс Карты)

ЗЫКОВ Тимур Алексеевич

*студент, Физтех-школа прикладной математики и информатики
Московского физико-технического института
(национального исследовательского университета)*

ИВАНОВ Виталий Витальевич

старший разработчик, ООО «Яндекс.Технологии» (Яндекс Маркет)

ЛЕВЕРЬЕВ Владимир Семенович

старший преподаватель кафедры ИТ ИМИ СВФУ

НИКИФОРОВ Дьулустан Васильевич

старший преподаватель кафедры ИТ ИМИ СВФУ

НИКОЛАЕВА Наталья Васильевна

*к. ф.-м. н., зав. кафедрой ИТ ИМИ СВФУ,
зав. кафедрой информатики МАН РС(Я)*

ОБУДОВ Владислав Антонович,

диспетчер Малой академии наук РС (Я) — секретарь жюри

ПАВЛОВ Александр Викторович

к. ф.-м. н., доцент кафедры ИТ ИМИ СВФУ

ЭВЕРСТОВ Владимир Васильевич

старший преподаватель кафедры ИТ ИМИ СВФУ

СЕМЕНОВ Никита Сергеевич

*студент, факультет компьютерных наук
Национального исследовательского университета
«Высшая школа экономики»*

УСЛОВИЯ ЗАДАЧ

А. Успеть к финалу

Бы́лый очень очень любит футбол и сегодня с нетерпением ждет дома трансляцию финала Лиги чемпионов.

Но вдруг Бы́лый вспомнил, что сегодня он еще должен сплавить на лодке за отцом, который ждет его в охотничьей избушке. Избушка находится прямо на берегу реки выше по течению на расстоянии L км от причала. Известно, что скорость лодки равна v_b км/ч, скорость течения реки — v_r км/ч. Временем, затрачиваемым на все сопутствующие дела, пренебрежём, так как Бы́лый живет у самого причала и лодка всегда наготове.

Ответьте на вопрос: успеет ли Бы́лый на финал Лиги чемпионов, если он выходит прямо сейчас, а финал начнется ровно через a минут?

Формат входных данных

На вход вашей программе дается четыре целых числа a, v_b, v_r, L ($0 < a, v_b, v_r, L \leq 10^6$).

Формат выходных данных

Ваша программа должна вывести строку «YES», если Бы́лый успеет на финал, и «NO» иначе.

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
320 20 5 50	YES
100 20 7 100	NO

Комментарии

Два числа с плавающей точкой будем называть равными друг другу в том случае, если модуль их разности не превышает 0.001.

В. Новое слово в криптографии

Начинающий программист Даринка с недавних пор увлеклась криптографией и понимает, что в основе шифрования сигналов лежат побитовые операции.

Для решения новой очень важной задачи шифрования ей нужно найти результат **побитового исключающего или** всех чисел от A до B , но у неё это совсем не получается.

Может, вы сможете ей помочь?

Формат входных данных

В единственной строке заданы два целых числа A и B ($0 \leq A \leq B \leq 10^{18}$).

Формат выходных данных

Вывести одно целое число — ответ на задачу.

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
1 19	0
1 2022	2023

Комментарии

Исключающее ИЛИ (XOR) устанавливает значение бита результата в 1, если значения в соответствующих битах исходных переменных различны:

$$1001 \text{ XOR } 0011 = 1010.$$

С. Калькулятор пиццы

Молодой программист Айтал получил заказ от сайта «Каталог калькуляторов» на разработку нового калькулятора пиццы. Пусть пиццерия предлагает пиццы трех разных размеров, известны их диаметры и цены, а покупатель желает приобрести несколько пицц общей площадью не менее N см² и при этом сэкономить.

Калькулятор должен определить минимальную сумму денег, которую придется потратить покупателю.

Формат входных данных

В первой строке задано целое число N — желаемая покупателем суммарная площадь пицц ($1 \leq N \leq 10^5$).

В трех следующих строках через пробел заданы по два целых числа: соответственно, диаметр d_i и цена p_i каждой i -й пиццы ($1 \leq d_i, p_i \leq 10^5$, $i = 1, 2, 3$).

Формат выходных данных

Программа должна вывести одно целое число — ответ на задачу.

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
1000 25 545 30 829 35 999	1374

Комментарии

В примере нужно купить одну среднюю и одну маленькую пиццу. Их суммарная площадь будет равна

$$\pi(25/2)^2 + \pi(30/2)^2 \approx 490.87 + 706.86 = 1197.73 > 1000.$$

Их общая стоимость: $545 + 829 = 1374$.

D. Армия Дыырая

В армии Дыырая четное количество воинов. Каждый воин имеет номер от 1 до N . Сила воинов увеличивается в два раза, если они сражаются парами и сумма номеров воинов — простое число.

Требуется написать программу, распределяющую воинов на максимально возможное количество таких пар.

Формат входных данных

Задано одно целое число N ($2 \leq N \leq 10^4$).

Формат выходных данных

Вывести список найденных пар. Каждая пара выводится в одной строке через пробел.

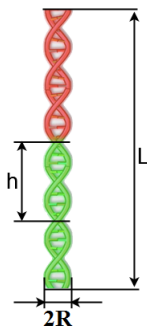
Пример входного и выходного файлов

стандартный ввод	стандартный вывод
8	3 8 4 7 5 6 1 2

Е. Нить ДНК

Тобул серьезно увлекается биологией, и совсем недавно он узнал, что все живые организмы имеют ДНК — молекулу-носитель генетической информации. Она состоит из двух цепей, каждая из которых закручена по *цилиндрической винтовой линии*.

Оказывается, у каждого живого организма молекула ДНК имеет свои параметры — своя длина (L), радиус витка нити ДНК (R), длина шага витка (h).



Тобула очень заинтересовал один вопрос: можно ли определить длину нити ДНК для определенного типа живого организма? Поскольку он все свое время посвящает биологии, то с математикой, к его большому стыду, у него не очень хорошо. Вот почему он просит вас помочь ему вычислить длину одной нити ДНК живого организма.

Формат входных данных

На вход вашей программе даются характеристики ДНК живого организма — три вещественных числа L , R , h , ($0 < L, R, h \leq 10^4$). Числа заданы с точностью до двух знаков после десятичной точки.

Формат выходных данных

Ваша программа должна вывести длину одной нити ДНК с точностью до четвертого знака после точки.

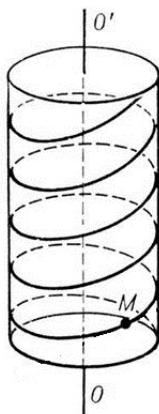
Пример входного и выходного файлов

стандартный ввод	стандартный вывод
10.00 3.00 4.00	48.1732

Комментарии

В ваших вычислениях значение π примите равным 3.141593.

Винтовая линия — это пространственная кривая, описываемая точкой M , которая вращается с постоянной угловой скоростью вокруг неподвижной оси OO' и одновременно перемещается вверх с постоянной скоростью вдоль этой оси.

**Ф. Рецепт пельменей**

Успешный спортивный программист Айтал не смог пройти собеседование в IT-компанию «Тындекс». Чтобы хоть как-то платить за подписку на стриминговый сервис «Кекфликс», он устроился на работу кассиром в магазин «Четвёрочка».

У «Кекфликс» очень дорогая подписка, а зарплаты кассира не хватает. Поэтому Айтал решил вечерами подрабатывать таксистом в компании «outDriver». Как-то раз Айтал взял заказ у CEO «outDriver» — Айсена Момского. Айтал, не зная кого везет, начал рьяно ругать приложение.

Айсен Момский и так был не в духе, а слова «А вот у Тындекса приложение куда лучше» окончательно лишили Айтала работы.

Расстроившись, Айтал пошёл домой смотреть сериалы. По пути домой он решил рискнуть и купить пельмени у конкурента Четвёрочки — магазина «Магнат». Но разведывательное управление Четвёрочки не дремлет — его тут же уволили. Так Айтал стал безработным.

Вскоре закончилась подписка на «Кекфликс», а от Айтала ушла девушка. Лишившись работы, девушки и сериалов, Айтал начал писать рэп.

Жюри не хочет, чтобы спортивные программисты становилось рэперами, поэтому дает вам для подготовки задачу, которую Айтал не смог решить на собеседовании.

Дан массив длины n , пронумерованный от 1 до n , изначально состоящий из нулей, и m запросов. В каждом запросе дано число x . Запросы имеют два типа:

- Ультралевые — среди всех элементов массива с индексами от 1 до x выбирается наименьший. Если таких несколько, то выбирается элемент с **наименьшим** индексом;
- Ультраправые — среди всех элементов массива с индексами от x до n выбирается наименьший. Если таких несколько, то выбирается элемент с **наибольшим** индексом.

Выбранный элемент увеличивается на один.

Требуется вывести полученный массив.

Формат входных данных

В первой строке заданы три целых числа n и m ($1 \leq n, m \leq 2 \cdot 10^5$) — длина массива и количество запросов.

В следующих m строках даны индекс x ($1 \leq x \leq n$) и код c — описание запроса. Код равен «l», если запрос ультралевый, и «r», если запрос ультраправый.

Формат выходных данных

Выведите n целых чисел — полученный массив.

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
5 4 3 1 1 r 4 1 2 1	2 1 0 0 1
5 10 5 1 5 1 1 r 5 r 4 r 3 1 3 r 1 1 1 r 5 1	2 2 2 2 2

Г. Своя игра

После очередной победы в тяжёлой игре, Ки Хун решил придумать свою игру. Участники игры должны пройти некоторый лабиринт от входа до выхода, собрав карточки у всех солдат, находящихся в лабиринте. Солдат отдаёт игроку свою карточку только в том случае, если игрок подошел к нему по *верному* маршруту. Маршрут от входа до выхода будет верным, если при условии обхода всех солдат, он будет наикратчайшим из всех возможных маршрутов.

Ки Хун дал вам протестировать эту игру и просит написать программу, которая для произвольного лабиринта выведет количество солдат в нем и длину верного маршрута до каждого очередного солдата.

Формат входных данных

В первой строке записаны натуральные числа N и M — размеры лабиринта ($3 \leq N, M \leq 10$).

Во второй строке записаны i_s и j_s — координаты входа в лабиринт,

в третьей строке записаны числа i_e и j_e — координаты выхода из лабиринта ($0 \leq i_s, i_e < N$, $0 \leq j_s, j_e < M$).

В последующих N строках через пробел записаны по M чисел, описывающих лабиринт: 0 — свободная клетка, 1 — стена, 2 — солдат.

Солдат в лабиринте может быть не более 10.

Формат выходных данных

В первой строке вывести искомое количество солдат K . Затем вывести K чисел — количество клеток лабиринта, пройденного до каждого очередного солдата по верному маршруту.

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
5 5 0 0 3 4 0 1 0 0 0 0 1 0 1 2 2 0 0 1 1 0 1 2 0 0 0 1 1 1 1	3 2 9 15

Комментарии

Гарантируется, что каждый солдат достижим для участников игры, и в лабиринте обязательно есть хотя бы один солдат. Также гарантируется, что идя по кратчайшему маршруту нельзя встретить одного и того же солдата более одного раза, а также гарантируется то, что верный маршрут — единственный.

В примере солдаты имеют следующие координаты: (2, 0), (3, 2) и (1, 4).

Н. Защита лилипутов

Населению Лилипутии с давних пор угрожают титаны-лилипутоеды. Для защиты от них лилипуты возводят большие стены, имеющие вид окружностей, некоторые из которых могут размещаться внутри других. Уже практически целый век лилипуты живут спокойно и даже начали забывать о беспощадных врагах. Но с приходом Колоссальных ти-

танов, способных разрушать стены, лилипутское общество вновь оказалось в опасности.

Известно, что в Лилипутии есть n стен, и любые две стены либо не пересекаются, либо вложены друг в друга. Колоссальных титанов оказалось так много, что в некоторый момент времени они могут сломать все внешние стены одновременно, и тогда лилипуты будут вынуждены отступить под защиту более внутренних стен.

Время на исходе, лилипуты паникуют. Король лилипутов обратился к вам за помощью, чтобы решить, куда лучше всего бежать. Помогите ему найти стену, которая падет последней.

Формат входных данных

В первой строке задано натуральное число n ($1 \leq n \leq 2 \cdot 10^3$).

В оставшихся n строках содержатся описания стен-окружностей — координаты центра (x_i, y_i) и радиус r_i каждой i -й окружности ($i = 1, \dots, n$, $1 \leq r_i, x_i, y_i \leq 10^8$, $0 \leq x_i - r, y_i - r \leq 10^8$).

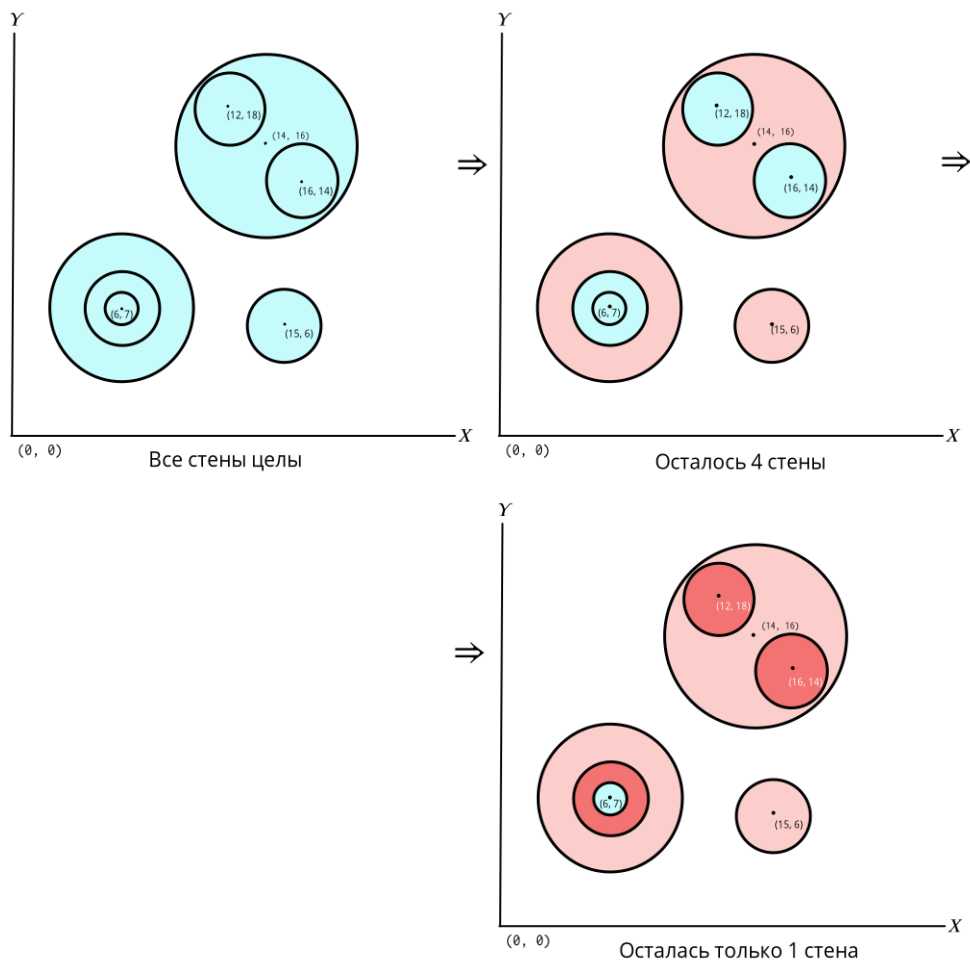
Формат выходных данных

Выведите порядковый номер самой внутренней стены (нумерация стен начинается с 1). Если ответов несколько, выведите любой из них.

Примеры входного и выходного файлов

стандартный ввод	стандартный вывод
2 2 2 1 2 2 2	1
7 6 7 2 6 7 5 15 6 2 6 7 1 14 16 5 16 14 2 12 18 2	4

Комментарии



На рисунках изменением цвета показано, в каком порядке как будут рушиться стены во втором примере. Последней падет стена под номером 4 с радиусом 5 и центром $(6, 7)$.

I. Контейнеры

Автоматический кран загружает контейнеры в трюм корабля через круглый люк диаметром d .

Все контейнеры имеют ширину a_i и длину b_i . Контейнеры можно поворачивать в горизонтальной плоскости.

Определите, сколько из имеющихся N контейнеров может быть загружено в трюм.

Формат входных данных

В первой строке даны целые положительные числа N и d ($1 \leq N, d \leq 10^4$).

В следующих N строках даны целые положительные числа a_i и b_i ($1 \leq a_i, b_i \leq 10^5$).

Разделителями во всех строках являются пробелы.

Формат выходных данных

Ваша программа должна вывести целое положительное число — ответ на задачу.

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
3 8 6 3 4 9 7 7	1

I. Дроби

Алтан учится в школе и начал изучать обыкновенные дроби. Сегодня на уроке математики всем ученикам учитель раздал N карточек с дробями:

$$\frac{x_1}{y_1}, \frac{x_2}{y_2}, \frac{x_3}{y_3}, \dots, \frac{x_i}{y_i}, \dots, \frac{x_N}{y_N}.$$

Записав на доске дробь $\frac{a}{b}$, учитель дал задание на дом — к следующему уроку принести 2 карточки, которые в сумме равны дроби $\frac{a}{b}$.

У Алтана не получается справиться с заданием, и поэтому он просит вас помочь решить эту задачу.

Формат входных данных

В первой строке находятся 3 целых числа N , a , b ($2 \leq N \leq 10^5$, $0 \leq |a|, |b| \leq 10^{17}$).

Каждая из последующих N строк содержит по 2 целых числа x_i, y_i ($0 \leq |x_i|, |y_i| \leq 10^8, i = 1, \dots, n$).

Гарантируется, что $b, y_i \neq 0$.

Формат выходных данных

Если ответом являются дроби $\frac{d}{e}, \frac{f}{g}$ — выведите 4 числа d, e, f, g . Если верных ответов несколько, то выведите любой из них.

Если ответа нет — выведите -1 .

Пример входного и выходного файлов

стандартный ввод	стандартный вывод
3 2 3	1 3 1 3
5 6	
1 3	
1 3	

РЕШЕНИЯ

А. Успеть к финалу

Время, необходимое для того, чтобы сплавить за отцом и вернуться домой, можно вычислить по формуле:

$$T = \frac{L}{v_b - v_r} + \frac{L}{v_b + v_r}.$$

Но в этом случае время будет в часах, а время до финала задано в минутах. Поэтому T нужно перевести в минуты — умножить его на 60, тогда получим:

$$T = 60 \cdot \frac{L}{v_b - v_r} + 60 \cdot \frac{L}{v_b + v_r}.$$

Бы́тый успеет, если $a \leq T$, но по условию задачи два числа с плавающей точкой равны между собой, если модуль разности этих двух чисел не превышает заданного ϵ . Тогда условие того, что Бы́тый успеет к началу матча будет выглядеть следующим образом:

$$T < a \text{ or } \text{abs}(T - a) \leq \epsilon.$$

В. Новое слово в криптографии

Решение задачи основывается на следующем факте « $A \text{ XOR } B \text{ XOR } B = A$ ». Тогда для нахождения XOR всех чисел от A до B можно воспользоваться следующей формулой:

$$\begin{aligned} & (A) \text{ XOR } (A + 1) \text{ XOR } (A + 2) \text{ XOR } \dots \text{ XOR } (B) = \\ & ((0) \text{ XOR } (1) \text{ XOR } (2) \text{ XOR } \dots \text{ XOR } (A - 1)) \text{ XOR} \\ & ((0) \text{ XOR } (1) \text{ XOR } (2) \text{ XOR } \dots \text{ XOR } (B)). \end{aligned}$$

XOR всех чисел от 0 до X , в свою очередь, достаточно легко выводится. Для этого, например, выпишите первые 16 двоичных чисел и посмотрите, что будет, если применить операцию XOR для них всех. Легко заметить, что начиная с 0, XOR каждой следующей пары чисел дает двоичную единицу ($0 \text{ XOR } 1 = 1$, $10 \text{ XOR } 11 = 1$ и т.д.). А далее каждые две единицы «нейтрализуют» друг друга. Поэтому рассматриваем остаток от деления X на 4:

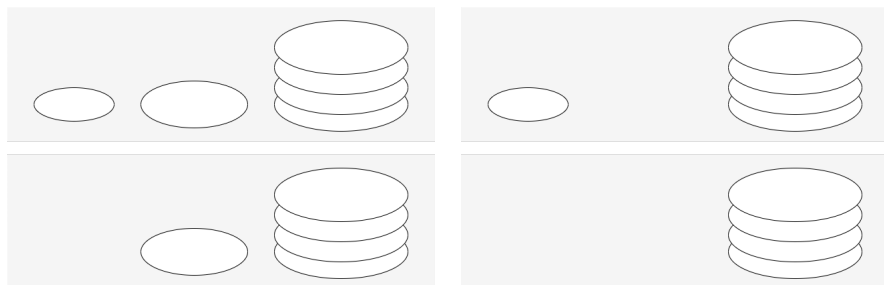
- если остаток равен 0, то XOR всех чисел до него самого равен 0, и ответом является число X ;
- если остаток равен 1, то XOR всех чисел вместе с ним равен 1;
- если остаток равен 2, то XOR всех чисел до него самого равен 1, и ответом является сумма 1 и числа X ;
- если остаток равен 3, то XOR всех чисел вместе с ним равен 0.

Определив функцию нахождения XOR всех чисел от 0 до X , далее остается только получить результат по формуле $\text{XOR}(\max(A - 1, 0)) \wedge \text{XOR}(B)$, где \wedge — операция XOR, определенная во многих языках программирования.

С. Калькулятор пиццы

Для решения задачи нужно набрать максимальное количество пицц одного размера и, возможно, по одной пицце других размеров.

Алгоритм должен выбрать в цикле максимизируемую пиццу, а затем вычислить стоимость 4 вариантов для двух пицц других размеров:



Вычислив стоимость этих 12 вариантов, нужно вывести наименьшую из них.

Д. Армия Дырая

Докажем, что мы всегда можем разбить воинов на $\frac{n}{2}$ простых пар. Сделаем это по индукции.

Для $n = 2$ разбиение очевидно и единственно: $1 + 2 = 3$.

Допустим, что разбиение на простые пары существует для всех n от 2 до $2(k-1)$. Докажем, что оно существует и для $n = 2k$. Приведем такое разбиение.

По теореме *Бертрана — Чебышёва* для любого натурального $n \geq 2$ найдётся простое число p в интервале $n < p < 2n$. Значит найдется такое m ($1 \leq m < 2k$), что $2k + m$ — простое ($2k < 2k + m < 4k$), причем m — нечетное. По предположению индукции разбиение чисел от 1 до $m-1$ существует. Числа от m до $2k$ будем разбивать следующим образом:

$$m + 2k, (m + 1) + (2k - 1), \dots, \frac{2k + m - 1}{2} + \frac{2k + m + 1}{2}.$$

Значит, для любого четного n , числа от 1 до n можно разбить на $\frac{n}{2}$ простых пар.

При решении задачи достаточно найти наибольшее m , такое, что $m + n$ простое, вывести разбиение чисел от m до n , приведенное в доказательстве, и дальше решать задачу для $n = m - 1$.

Ниже приведено решение задачи на языке Python:

```
def is_prime(x):
    d = 2
    while d*d <= x:
        if x % d == 0:
            return False
        d += 1
    return True

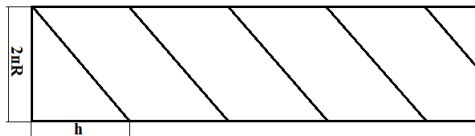
def print_primes(l, r):
    while l < r:
        print(l, r)
        l += 1
        r -= 1

n = int(input())

while n > 0:
    x = n - 1
    while not is_prime(x + n):
        x -= 1
    print_primes(x, n)
    n = x - 1
```

Е. Нить ДНК

Задача на первый взгляд является пространственной, но ее очень легко перевести на плоскость. Для этого разрежем цилиндр параллельно оси OO' и развернем его, тогда цилиндр превращается в прямоугольник. Так как точка M вращается вокруг оси OO' равномерно и движется вверх тоже равномерно, то на плоскости прямоугольника винтовая линия будет представлена множеством параллельных отрезков, суммарную длину которых и требуется найти.



Длину одного отрезка легко найти, используя теорему Пифагора:

$$L_1 = \sqrt{4\pi^2 R^2 + h^2}.$$

Тогда длина одного витка ДНК может быть вычислена следующим образом:

$$L_{DNA} = L_1 \frac{L}{h}.$$

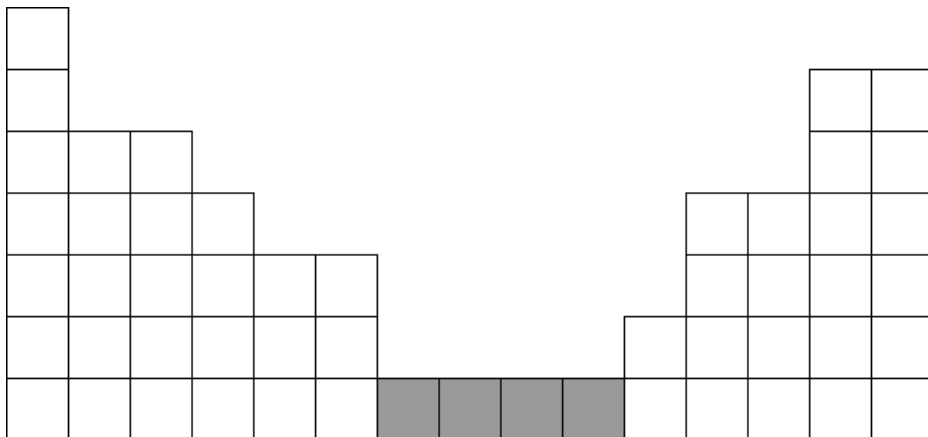
Г. Рецепт пельменей

По условию задачи нам нужно уметь находить первый минимум на отрезке от 1 до x и последний минимум на отрезке от x до n , и затем увеличивать найденный элемент на 1. Это стандартная задача *RMQ* (*Range minimum query*). Ограничения позволяют решать эту задачу за время $O(n \log n)$ с использованием, например, *дерева отрезков*.

Но данная задача имеет также линейное решение (за $O(n)$). Так как мы находим **первый** минимум на **префиксе** и **последний** минимум на **суффиксе**, то наш массив всегда имеет следующий вид:

- слева — невозрастающий префикс (возможно нулевой длины);
- справа — неубывающий суффикс (возможно нулевой длины);

- между ними — подотрезок ненулевой длины, состоящий из одинаковых минимальных элементов массива. Далее будем называть его *дном* (*bottom*). $bottom_l$ — индекс левого края дна, $bottom_r$ — индекс правого края дна.



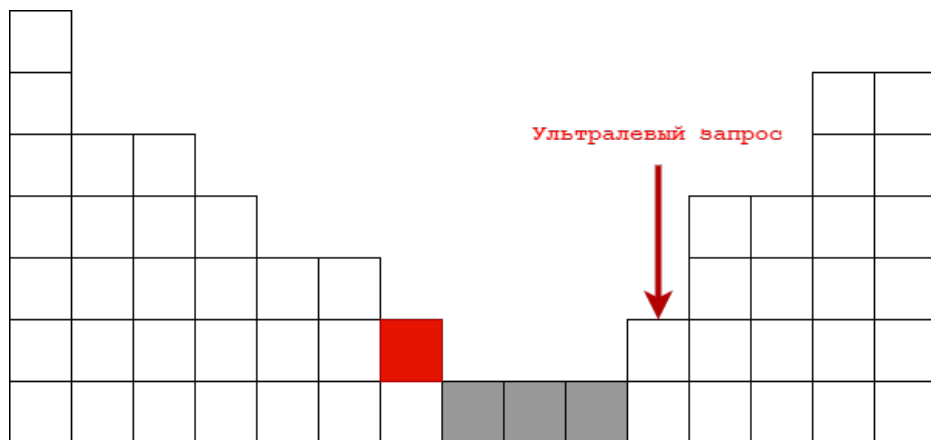
Для массива $[7, 5, 5, 4, 3, 3, 1, 1, 1, 1, 2, 4, 4, 6]$. Серым обозначено дно

Обозначим массив как a . При запросе x 1:

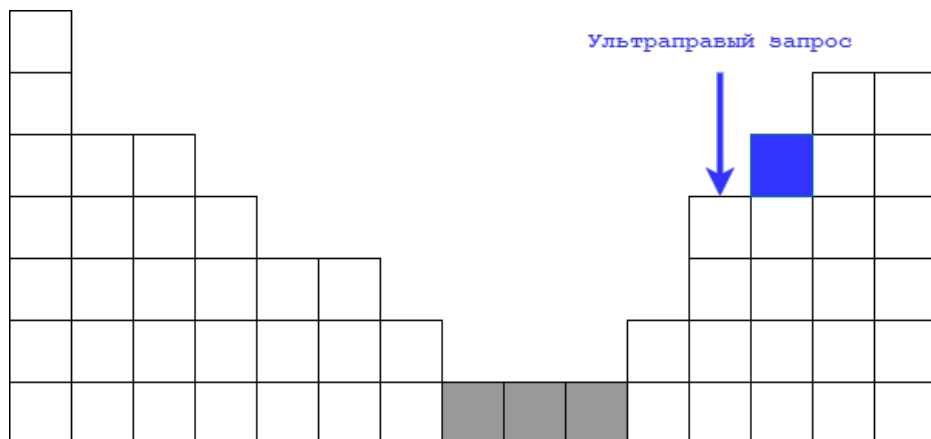
- Если $x \geq bottom_l$, то увеличивается элемент $a[bottom_l]$, а $bottom_l$ увеличивается на 1;
- Если $x < bottom_l$, то увеличивается самый левый элемент, равный $a[x]$ в подотрезке из элементов $a[x]$ в невозрастающем префиксе. В нём подотрезок из элементов $a[x]$ укорачивается слева, а подотрезок из элементов $a[x] + 1$ увеличивается справа.

Аналогично при запросе x r:

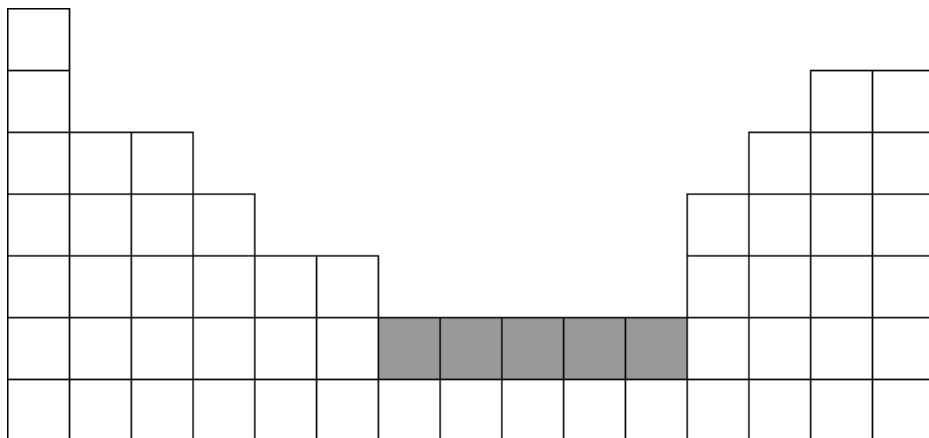
- Если $x \leq bottom_r$, то увеличивается элемент $a[bottom_r]$, а $bottom_r$ уменьшается на 1;
- Если $x > bottom_r$, то увеличивается самый правый элемент, равный $a[x]$ в подотрезке из элементов $a[x]$ в невозрастающем суффиксе. В нём подотрезок из элементов $a[x]$ укорачивается справа, а подотрезок из элементов $a[x] + 1$ увеличивается слева.



После запроса 11 l



После запроса 12 r



Новое дно после трех запросов 8 r

Промоделируем задачу. Будем поддерживать правую и левую границы подотрезков равных элементов в невозрастающем префиксе и неубывающем суффиксе. При увеличении элемента $a[x]$ соответствующая граница подотрезка с элементами $a[x]$ уменьшается, а соответствующая граница подотрезка с элементами $a[x] + 1$ увеличивается. Когда подотрезки сольются, обновляем дно.

Г. Своя игра

Для решения задачи необходимо:

1. Для стартовой и конечной точки, а также каждого солдата в лабиринте следует найти кратчайшие расстояния между ними и запомнить их в удобную структуру для хранения такого графа (например, матрицу смежности). Наиболее эффективным способом поиска таких расстояний будет использование алгоритма поиска в ширину на двумерном массиве (BFS).
2. Выбор наилучшего маршрута в полученном полном графе следует делать через полный перебор с помощью метода перестановок. В данном случае объектом перебора будет последовательность солдат. То есть необходимо выбрать минимальную сумму расстояний от стартовой точки до конечной через всех солдат.

Н. Защита лилипутов

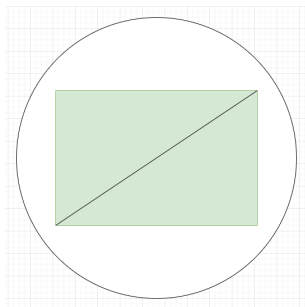
Попробуем оценить для каждого круга степень его вхождения в другие. Отсортируем круги по невозрастанию радиуса. Пусть для первых k кругов уже знаем степень вложенности. Тогда степень вложенности $(k + 1)$ -го круга — на один больше, чем максимальное значение степени вложенности среди первых k кругов, внутри которого находится $(k + 1)$ -й круг. Ответ предыдущих не будет модифицироваться, так как круги отсортированы по невозрастанию радиуса. База очевидна — самые большие круги имеют степень вложенности 0. Асимптотика алгоритма — $O(N^2)$.

Псевдокод:

```
1  vector<Walls> walls;
2  int dp[10005]= {0,..., 0};
3  in >> walls;
4  sort(walls, compareByRadius);
5  for(int i = 0; i < wallsCount; ++i) {
6      for(int j = 0; j < i; ++j) {
7          if(walls[i].isContainedIn(walls[j])) {
8              dp[i] = max(dp[i], dp[j] + 1);
9          }
10     }
11 }
12 out << getIndex(max(dp));
```

I. Контейнеры

Для каждого контейнера с помощью теоремы Пифагора вычисляем диагональ контейнера.



Контейнер помещается в круглый люк, если диагональ контейнера меньше, чем его диаметр.

II. Дроби

Основная проблема данной задачи в том, что дроби могут быть приведены в виде сократимой дроби. Для решения этой проблемы нужно уметь приводить их к нормальной несократимой форме (числитель и знаменатель взаимно простые и знаменатель больше нуля). Для приведения дробей к нормальной форме нужно поделить числитель и знаменатель на их наибольший общий делитель.

Для вычисления суммы/разности дробей нужно привести их к общему знаменателю и сложить/вычесть числители. Полученную дробь также нужно нормализовать.

Перейдем к решению самой задачи. Заведем словарь — ключом будет выступать нормализованная дробь, а значением будет представление дроби во входных данных. Проходим по всем карточкам с дробями. Для каждой дроби выполняем следующие действия:

- вычисляем разность требуемой дроби $\frac{a}{b}$ и $\frac{x_i}{y_i}$;
- проверяем наличие этой разности в словаре, если дробь нашлась — ответом являются дроби $\frac{x_i}{y_i}$ и значение в словаре, иначе вычисляем нормализованную дробь и записываем ее в словарь.

ПРОТОКОЛ

ХІХ Республіканська командна олімпіада школярів по програмуванню

№	Команда	A	B	C	D	E	F	G	H	I	J	=	Время	Место	Диплом
1	ЕРЭЛ1 (Парников В. В., Саввинов О. С., Власов Я. А. – 11 кл.)	+2 00:22	+ 00:14	+1 04:54	+ 02:32	+ 00:29	+ 01:43	+ 01:12	+1 01:19	+ 00:25	+6 04:41	10	1271	1	I ст.
2	ФТЛ-1 (Захаров А. А., Белюсов И. А., Павловский В. В. – 10 кл.)	+ 00:13	+ 04:47	+1 03:18	+ 02:15	+ 00:47			+1 01:32	+ 00:18	-17	7	830	2	II ст.
3	СПЛ-1 (Петров Т. Р. 10 кл., Кузнецов Ю. Д. 11 кл., Туприн А. К. 11 кл.)	+3 00:20	+3 02:09		+ 03:34	+ 00:36	-1		+1 04:10	+ 01:08	+5 04:59	7	1256	3	II ст.
4	ЕРЭЛ2 (Кузьмин А. А., Уваровская Л. А., Павлов Н. М. – 11 кл.)	+4 00:26	+ 02:01	-5	-1	+ 00:52	+ 03:10		+ 01:12	+ 00:35	-5	6	576		II ст.
5	ЕРЭЛ3 (Ефимов С. С., Миндаев Д. А., Шумилов А. К. – 10 кл.)	+3 00:26	+ 04:37		+1 03:23	+ 01:08			-3	+ 01:14		5	728		III ст.
6	ЕРЭЛ5 (Семенов Е. Н., Чичигинаров В. К., Харлампьев Р. И. – 10 кл.)	+ 00:15			-1	+ 02:04	-3		+ 04:25	+3 01:35	-3	4	559		III ст.
7	ЕРЭЛ4 (Фролова К. Г. 10 кл., Макаров Д. И. 11 кл., Акимов М. М. 10 кл.)	+4 01:20			+1 02:46	+1 04:00				+ 01:41		4	707		III ст.
8	ЕРЭЛ6 (Павлова А. Е. 8 кл., Решетников Н. П. 8 кл., Юмшанов Э. П. 7 кл.)	+ 00:42	-5			+ 00:34				+ 00:48	-3	3	124		Похв. грамота

9	СПТЛИ (Спиридонов Э. С., Мекумянов Д. Л., Федоров Т. В. – 10 кл.)	+1 00:30	-1		-3	+ 01:12				+ 01:28	3	210	Похв. грамо- та
10	СПЛ-2 (Федоров А. А. 8 кл., Гаврильев А. В. 9 кл., Давлетов А. 9 кл.)	+1 00:31		-3		+ 01:27	-3			+ 01:46	3	244	Похв. грамо- та
11	АКБ (Макаров М. Е., Корогов В. В., Гаврильев Д. Г. – 9 кл.)	+4 00:49	-2	-14	-5	+ 01:26				+ 01:36	3	311	Похв. грамо- та
12	Чурапча (Куличкин Т. Ф. 10 кл., Корякин М. Н. 10 кл., Барашков В. В. 11 кл.)	+6 01:27				+ 01:17	-3		-2	+ 00:33	3	317	Похв. грамо- та
13	Нюрба - НТЛ (Захаров В. В. 11 кл., Кириллин М. Р. 11 кл., Федоров А. В. 8 кл.)	+4 01:01	-3	-3		+ 03:32				+ 00:38	3	391	Похв. грамо- та
14	ФТЛ-6 (Чабыкин О. А. 10 кл., Тихомиров Д. 10 кл., Герасимов А. А. 10 кл.)	+5 01:06	-1		-1	+ 02:47	-1			+ 01:14	3	407	Похв. грамо- та
15	ЭРЭЛ7 (Ноговицын А. Е., Шамаев Б. Э., Николаев Д. А. – 9 кл.)	+16 02:44			+ 01:47	-4	-1			+ 02:10	3	721	Похв. грамо- та
16	ERSEVL (Степанов С. Г. 11 кл., Филиппов В. А. 10 кл., Винокуров Э. Р. 9 кл.)	+ 00:32	-5						-1	+ 00:36	2	68	

17	СОШ17 (Солопов М.Д. 11 кл., Курчатов А.Н. 10 кл., Семенов А.Г. 9 кл.)	+ 00:26	-1		-1	-2			+ 01:12	-1	2	98	
18	Siience (Ефимов В.М., Слепцов В. В., Загоренко А.А. – 10 кл.)	+3 00:33	-3						+ 02:13	-1	2	226	
19	NoNames (Егоров Е.Г., Иванов П.А., Николаев М.А. – 8 кл.)	+ 00:16	-9		-2				+6 03:08		2	324	
20	MLT (Бурнашев М.В., Егорова Л. М., Петров Н.И. – 9 кл.)	+5 02:57	-2		-1				+ 02:18	-1	2	415	
21	ПСОШ№3 (Панов С.В. 9 кл., Гуськов Д.А. 9 кл., Капитонов А.В. 11 кл.)	+6 03:07	-1								1	307	
22	ФТЛ-7 (Елизаров В.П. 5 кл., Гагарин Я.И. 6 кл., Потапов К. 5 кл.)	+11 03:44	-1	-19					-2		1	444	
23	СОШ 12 (Николаев В.А., Максимов Д.А., Аргунов А.А. – 9 кл.)	-2	-1			-2					0	0	
24	ФТЛ-5 (Местников М. П., Привалов В.Д. – 9 кл.)	+2 00:26	-1	-3	+ 01:43				+ 01:52	-6	3	281	вне заче- та
25	ФТЛ-2 (Кривошапкин Э.А., Дмитриева А.С. – 8 кл.)	+10 03:33	-1	-11	+ 04:33				+6 03:55	-6	3	1041	вне заче- та

Председатель жюри: Н.Н. Павлов

5 марта 2022 г.

Содержание

Условия задач	6
A. Успеть к финалу	6
B. Новое слово в криптографии	6
C. Калькулятор пиццы	7
D. Армия Дыырая	8
E. Нить ДНК	9
F. Рецепт пельменей	10
G. Своя игра	12
H. Защита лилипутов	13
I. Контейнеры	15
J. Дробь	16
Решения	18
A. Успеть к финалу	18
B. Новое слово в криптографии	18
C. Калькулятор пиццы	19
D. Армия Дыырая	19
E. Нить ДНК	21
F. Рецепт пельменей	21
G. Своя игра	24
H. Защита лилипутов	25
I. Контейнеры	26
J. Дробь	26
Итоги олимпиады	27