Олимпиада «Ломоносов» по информатике. 2024-2025 учебный год
Заключительный этап. 7-10 классы
Задачи

Задача 1. Незнайка и Коллатц

Условия задачи 1

Незнайка очень полюбил функцию Коллатца $f(n) = \begin{cases} \frac{n}{2}, & n - \text{четное число} \\ 3n+1, & n - \text{нечетное число} \end{cases}$, про которую есть недоказанная гипотеза о том, что её применение снова и снова, стартуя с любого натурального ненулевого числа, всегда приводит к числу 1, после чего возникает цикл $1 \to 4 \to 2 \to 1$. Незнайке пришло в голову, что можно применять обратное преобразование, и от какого-то натурального ненулевого числа a переходить к одному из чисел b, такому что $f(b) = a$. Он обозначил такое обратное преобразование f^{-1} и стал искать способы перейти от стартового числа n_1 к финишному числу n_2 за минимально возможное количество шагов, применяя на каждом шаге либо функцию Коллатца f, либо обратное преобразование f^{-1} к текущему результату, полученному на предыдущем шаге (на первом шаге за текущий результат Незнайка брал число n_1). Например, от $n_1 = 10$ к $n_2 = 21$ можно перейти за 5 шагов: $10 \to f(10) = 5 \to f(5) = 16 \to f^{-1}(16) = 32 \to f^{-1}(32) = 64 \to f^{-1}(64) = 21$

Полагая, что гипотеза Коллатца верна, помогите Незнайке написать программу, которая определяет минимально возможное количество шагов, за которые из данного стартового числа n_1 можно получить финишное число n_2.

н *Формат ввода:* В первой строке вводится натуральное число n_1: $1 \le n_1 \le 5000$. Во второй строке содержится натуральное число n_2: $1 \le n_2 \le 5000$.

Формат вывода: В единственной строке выводится либо наименьшее из возможных количество последовательных применений функции $f(x)$ или преобразования $f^{-1}(x)$, после которых стартовое число n_1 преобразуется в финишное число n_2, либо -1, если стартовое число n_1 не может быть преобразовано таким способом в финишное число n_2. Если $n_1 = n_2$, то выводится 0.

Ввод примера №1:	Ввод примера №2:	Ввод примера №3:	Ввод примера №4:
10	8	5000	2
21	1	5000	1
Вывод примера №1:	Вывод примера №2:	Вывод примера №3:	Вывод примера №4:
5	2	0	1

Задача 2. Незнайка и недосортировка

Условия задачи 2

Знайка ввёл в Цветочном городе буквенную систему счисления. Это позиционная система счисления с основанием 26, в которой цифрами служат строчные латинские буквы и только они. В ней используется такая таблица цифр и их значений:

цифра	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
значение	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Приведем пример записи числа в буквенной системе счисления: $2025_{10} = 2{*}26^2 + 25{*}26^1 + 23{*}26^0 = czx$.

Все стали пользоваться буквенной системой счисления, как положено, кроме Незнайки. Однажды Знайка заглянул в исписанный Незнайкой листок, схватился за голову и пришёл в ужас. В записи чисел помимо обычных латинских букв Незнайка задействовал как цифры буквы с крышками ($\widehat{a} \dots \widehat{z}$), буквы с волной ($\widetilde{a} \dots \widetilde{z}$) и буквы с нижними подчёркиваниями ($\underline{a} \dots \underline{z}$). Знайка обратился к Незнайке за разъяснениями. Выяснилось, что Незнайка модифицировал буквенную систему счисления, разрешив переполнять в ней разряды, т. е. записывать в них цифры со значениями, превышающими 25. При этом основание системы счисления он оставил прежним – 26. Незнайка дополнил таблицу цифр и их значений:

цифра	$\widehat{a}$	$\widehat{b}$	$\widehat{c}$	$\widehat{d}$	$\widehat{e}$	$\widehat{f}$	$\widehat{g}$	$\widehat{h}$	$\widehat{i}$	$\widehat{j}$	$\widehat{k}$	$\widehat{l}$	$\widehat{m}$	$\widehat{n}$	$\widehat{o}$	$\widehat{p}$	$\widehat{q}$	$\widehat{r}$	$\widehat{s}$	$\widehat{t}$	$\widehat{u}$	$\widehat{v}$	$\widehat{w}$	$\widehat{x}$	$\widehat{y}$	$\widehat{z}$
значение	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
цифра	$\widetilde{a}$	$\widetilde{b}$	$\widetilde{c}$	$\widetilde{d}$	$\widetilde{e}$	$\widetilde{f}$	$\widetilde{g}$	$\widetilde{h}$	$\widetilde{i}$	$\widetilde{j}$	$\widetilde{k}$	$\widetilde{l}$	$\widetilde{m}$	$\widetilde{n}$	$\widetilde{o}$	$\widetilde{p}$	$\widetilde{q}$	$\widetilde{r}$	$\widetilde{s}$	$\widetilde{t}$	$\widetilde{u}$	$\widetilde{v}$	$\widetilde{w}$	$\widetilde{x}$	$\widetilde{y}$	$\widetilde{z}$
значение	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
цифра	$\underline{a}$	$\underline{b}$	$\underline{c}$	$\underline{d}$	$\underline{e}$	$\underline{f}$	$\underline{g}$	$\underline{h}$	$\underline{i}$	$\underline{j}$	$\underline{k}$	$\underline{l}$	$\underline{m}$	$\underline{n}$	$\underline{o}$	$\underline{p}$	$\underline{q}$	$\underline{r}$	$\underline{s}$	$\underline{t}$	$\underline{u}$	$\underline{v}$	$\underline{w}$	$\underline{x}$	$\underline{y}$	$\underline{z}$
значение	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103

Знайка сразу понял, что модифицированная система позволяет одно и то же число записать поразному. Например:
$2025_{10} = 2{*}26^2{+}25{*}26^1{+}23{*}26^0 = czx = 1{*}26^2{+}51{*}26^1{+}23{*}26^0 = b\widehat{z}x = 77{*}26^1{+}23{*}26^0 = \widetilde{z}x = 76{*}26^1{+}49{*}26^0 = \widetilde{y}\widehat{x} = 75{*}26^1{+}75{*}26^0 = \widetilde{x}\widetilde{x} = 74{*}26^1{+}101{*}26^0 = \widetilde{w}\underline{x} = 1{*}26^2{+}48{*}26^1{+}101{*}26^0 = b\widehat{w}\underline{x} = 2 * 26^2 + 22 * 26^1 + 101 * 26^0 = cw\underline{x}$.
Указаны некоторые, но не все способы записи числа 2025_{10}.

Знайка решил доказать Незнайке его неправоту. Он предложил Незнайке отсортировать по возрастанию последовательность различных чисел, записанных в модифицированной Незнайкой системе счисления. Знайка рассчитывал на то, что недостатки «переполненной» системы счисления станут очевидны сразу, как только понадобится выполнять сравнения чисел, записанных с её помощью. Но не тут-то было. Незнайка и бровью не повёл, и вскоре Знайка получил результат сортировки. Зная «творческий» подход Незнайки и заподозрив неладное, Знайка вгляделся в результат. И точно, сбылись его подозрения. Последовательность $A[i]$ оказалась недосортированной. В ней встретилась одна пара элементов $A[i_1]$ и $A[i_2]$ ($i_1 \neq i_2$), стоящих не на своих местах. Последовательность оказалась бы возрастающей лишь после обмена местами $A[i_1]$ и $A[i_2]$: установки прежнего числа $A[i_1]$ в позицию i_2 и прежнего числа $A[i_2]$ в позицию i_1.

Помогите Незнайке составить программу, находящую в недосортированной по возрастанию последовательности пару элементов, после обмена которых местами последовательность становится возрастающей. Программа считывает десятичное натуральное ненулевое число N, а затем последовательность из N записей чисел в модифицированной Незнайкой системе счисления. Во вводе буква с крышкой задаётся двумя символами, например, `a^` означает $\widehat{\mathtt{a}}$. Буква с волной также задаётся двумя символами, например, `b~` означает $\widetilde{\mathtt{b}}$. Буква с подчёркиванием также задаётся двумя символами, например, `c_` означает $\underline{\mathtt{c}}$. Заранее известно, что последовательность является недосортированной по возрастанию (в описанном выше смысле). Программа находит номера двух элементов, которые следует переставить, чтобы последовательность стала возрастающей, и выводит эти номера. Первым выводится меньший номер, вторым – больший. Нумерация элементов последовательности начинается с единицы.

Пример недосортированной по возрастанию последовательности из пяти элементов: `jw_ b d a kx~`. После обмена местами первого и четвертого элементов она становится отсортированной по возрастанию: `a b d jw_ kx~`

Формат ввода: В первой строке вводится натуральное число N в десятичной записи: $2 \leq N \leq 1000$. В последующих N строках содержатся записи различных неотрицательных чисел $A[i]$ ($i = 1..N, A[i] \neq A[j], i \neq j$) в системе счисления, модифицированной Незнайкой. В каждой записи используются только строчные латинские буквы, крышка `^`, волна `~`, подчёркивание `_`. Длина каждой записи $A[i]$ не более чем 25. В каждой строке содержится только запись одного числа.

Формат вывода: В единственной строке выводятся искомые номера i_1 и i_2 ($i_1 < i_2$), записанные в десятичной системе и разделённые одиночным пробелом.

```
Ввод примера №1:
5
jw_
b
d
a
kx~
Вывод примера №1:
1 4
Ввод примера №2:
4
mx
m~x~
m^x^
m_x_
Вывод примера №2:
2 3
Ввод примера №3:
2
bcdrfghijklmnopqrstuvwxzz
bcdrfghijklmnopqrstuvwxyz
Вывод примера №3:
1 2
```

Задача 3. Незнайка и рогейн

Условия задачи 3

Незнайка побывал в Австралии, где услышал про рогейн – вид спорта, изобретённый в этой стране. В рогейне от участника требуется быстрое передвижение бегом по пересечённой местности, навыки ориентирования с помощью карты и компаса, а также точное планирование своего маршрута. На карте соревнований по рогейну обозначены контрольные пункты, за посещение каждого из которых начисляется некоторое фиксированное количество очков. Разные пункты могут приносить разные количества очков, или одинаковые, – это решают устроители соревнований до начала рогейна. Каждый пункт участник может посетить столько раз, сколько пожелает, но очки начисляются только при первом посещении пункта. Цель каждого участника рогейна, набрать наибольшую сумму очков, затратив на передвижение между контрольными пунктами время, не превышающее максимальный предел, установленный организаторами.

Незнайка уже было решил заняться рогейном, но призадумался над тем, как он будет планировать свой маршрут. Поэтому Незнайка составил программу, которая поможет ему. Входные данные программы содержат сведения об N контрольных пунктах, у каждого из которых известно количество очков за его посещение – p_i. Маршрут Незнайки начинается в пункте 1. За время, не превышающее заданное время T, Незнайка должен посетить выбранные им контрольные пункты и успеть вернуться в пункт 1. Выбирать контрольные пункты для маршрута Незнайка должен так, чтобы набрать максимально возможную сумму очков. Про любые два контрольных пункта u и v Незнайка имеет сведения о том, сколько времени он потратит на передвижение от пункта u до пункта v. Программа находит набор контрольных пунктов, которые можно успеть посетить и набрать при этом наибольшую сумму очков. При выводе результата номера пунктов сортиру-

ются по возрастанию. Набор пунктов всегда начинается с 1. В наборе пункты не повторяются, даже если пункт при передвижении был посещён неоднократно, его номер в ответе выводится один раз. В частности, набор из более чем одного пункта заканчивается не на 1, так как никакой номер пункта не повторяется. Если программа находит несколько наборов пунктов, то она оставляет лишь те, которые имеют наименьший размер. Если после этого остаётся один вариант, то он и является ответом. Если остаётся несколько вариантов, то программа упорядочивает наборы пунктов лексикографически и считает ответом набор пунктов, оказавшийся на первом месте.
Составьте свою программу, дающую те же ответы, что и программа, составленная Незнайкой.
Рассмотрим один пример. Пусть имеется три пункта. Пусть передвижение между любой парой пунктов занимает 1 минуту. Пусть предельное время составляет 10 минут. Пусть количества очков за посещения пунктов равны 1, 2 и 3, соответственно. Программа найдёт два набора пунктов: 1) `1 2 3` и 2) `1 3 2`. После сортировки номеров пунктов по возрастанию оба набора запишутся одинаково: `1 2 3`. Эта запись и будет ответом.
Рассмотрим ещё один пример. Пусть, как и ранее, имеется три пункта. Пусть передвижение между первым и вторым пунктом занимает 5 минут. Пусть передвижение между первым и третьим пунктом занимает 5 минут. Пусть передвижение между вторым и третьим пунктом занимает 9 минут. Пусть предельное время составляет 10 минут. Пусть количества очков за посещения пунктов равны 1, 5 и 5, соответственно. В таких условиях есть два маршрута подходящей длительности, передвигаясь по которым удастся посетить два пункта, успеть вернуться в первый и заработать 6 очков: $1 \to 2 \to 1$ и $1 \to 3 \to 1$. Разных наборов пунктов у этих маршрутов будет два: 1) `1 2` и 2) `1 3`. Лексикографически первый набор предшествует второму. Он и будет ответом.

Формат ввода:
В первой строке вводятся два натуральных числа N T $(N \leq 10,\ T \leq 10^9)$ – количество контрольных пунктов, максимальный предел времени. Во второй строке даны N натуральных чисел – количества очков за посещение контрольных пунктов – $p_1, \ldots, p_N (p_i \leq 10^9)$. В следующих N строках дана таблица G размера $N \times N$ из натуральных чисел, где $G[i][j] = G[j][i] \leq 10^9$, и $G[i][j]$ задаёт время, которое потребуется для перемещения между пунктами i и j. Гарантируется, что $G[i][i] = 0$.

Формат вывода:
В первой строке программа выводит K – количество пунктов наборе, составленном так, как описано выше. Во второй строке программа выводит номера пунктов по возрастанию, разделяя одиночными пробелами, не ставя пробел после последнего пункта в наборе: $n_1\ n_2\ \ldots\ n_K$, где $n_1 = 1$ и $n_K \neq 1$ при $K \neq 1$.

```
Ввод примера №1:
3 10
1 4 3
0 1 1
1 0 1
1 1 0
Вывод примера №1:
3
1 2 3
Ввод примера №2:
3 10
1 5 4
0 5 5
5 0 9
5 9 0
Вывод примера №2:
2
1 2
```

```
Ввод примера №3:
3 10
1 4 5
0 9 9
9 0 9
9 9 0
Вывод примера №3:
1
1
```

Задача 4. Незнайка и химия

Условия задачи 4

Доктор Пилюлькин изготавливает в лаборатории разные лекарственные вещества. Доктор Пилюлькин почти знает какие компоненты нужны ему для приготовления лекарств, и какие вещества будут получаться в результате реакции. Свои знания Доктор записывает в виде формул химической реакции, которые могут быть неточны. Доктор попросил Незнайку исправить ему формулы для приготовления веществ.

Молекулы состоят из атомов. Атом — это либо одна заглавная латинская буква, либо заглавная латинская буква, за которой идут одна или более строчных латинских букв. Примеры обозначения атомов: H, El, Elerium. Если после обозначения атома записано положительное целое число, оно обозначает число атомов этого типа. Молекула состоит из атомов с указанием числа повторений. Примеры молекул: H2O, C2H5OLi. для группировки в записи молекулы могут использоваться круглые или квадратные скобки, но тогда число повторений относится ко всей группе атомов в скобках. Например, [(CH2)10HSi]2.

Формула реакции описывает превращение химических веществ. В процессе реакции одни молекулы превращаются в другие молекулы. Формула реакции записывается в виде I1+I2+I3...=O1+O2.... Молекулы с левой стороны от знака равенства — исходные вещества, а молекулы с правой части — результат реакции. Молекулы в левой и правой части разделяются знаком сложения. И в левой, и в правой части формулы реакции могут находится одна или более обозначений молекул. Перед обозначением молекулы может быть записано положительное целое число, обозначающее количество молекул в реакции. Пример реакции: 2H2+O2=2H2O.

В правильно записанной формуле должен выполняться закон сохранения вещества. Количество атомов каждого вещества в левой части формулы должно быть равно количеству атомов в правой части формулы реакции. Например, реакция H2+O2=H2O или реакция 3H2+CO2=H2O записаны неправильно.

Напишите программу, которая из возможно неправильной записи химической реакции получит правильную запись. Преобразованная запись химической реакции должна удовлетворять следующим требованиям:
∘ Молекулы веществ в левых и правых частях формулы должны следовать в порядке их первого вхождения в левую и правую части соответственно.
∘ Если одна и та же молекула входит в левую или правую части формулы несколько раз, все вхождения, кроме первого игнорируются. Запись молекулы для этого должна быть посимвольно совпадающей.
∘ Если в левой или правой части формулы полностью отсутствуют атомы, нужные в другой

части, недостающие атомы должны быть записаны в лексикографическом порядке в соответствующей части формулы после веществ из исходной формулы.
∘ Должен выполняться закон сохранения вещества.
∘ Числовые коэффициенты перед молекулами должны быть минимально возможными. Поэтому формула реакции 4H2+2O2=4H2O неправильная.

Формулы реакции вводятся на стандартном потоке ввода по одной формуле на строке. Исправленные формулы должны быть выведены на стандартный поток вывода по одной формуле на строке.
Для представления количества атомов в левой и правой частях формулы и для всех промежуточных вычислений достаточно положительных знаковых 32-битных целых чисел.

`Формат ввода:`
Формулы реакции вводятся на стандартном потоке ввода по одной формуле на строке. Количество атомов в левой и правой частях и входной формулы, и выходной формулы представимо положительным знаковым 32-битным целым числом. Длина одной входной строки не превышает 100000 байт. Суммарный размер всех входных строк не превышает 1000000 байт.

`Формат вывода:`
Исправленные формулы должны быть выведены на стандартный поток вывода по одной формуле на строке.

`Пример ввода:`

```
H2+O2=H2O
Si+O2=SiO2
C+O2+C+O2+C=CO2
(He[C2(O2Mg)4(O2Fe)2]2N3)2H3=(He(C2(O2Mg)4(O2Fe)2)2N3)2H3F4
```

`Пример вывода:`

```
2H2+O2=2H2O
Si+O2=SiO2
C+O2=CO2
(He[C2(O2Mg)4(O2Fe)2]2N3)2H3+4F=(He(C2(O2Mg)4(O2Fe)2)2N3)2H3F4
```

Задача 5. Незнайка и эпидемия

Критерий оценивания решений задачи 5
Общим критерием оценивания решения любой задачи является количество тестов, верно пройденных программой, составленной участником. Сначала решение проверяется на тестах из условия. Если не пройден хотя бы один из них, то решение оценивается в 0 технических баллов. Если тесты из условия пройдены, то решение проверяется на основном наборе тестов. Высчитывается доля (в процентах) верно пройденных тестов из основного набора по отношению к общему количеству тестов в наборе. Полученное количество процентов, округлённое до целого, определяет количество технических баллов за решение.

Технические баллы за задачу 5
За решение задачи начисляется от 0 до 100 технических баллов по критерию, указанному выше.

Перевод технических баллов в оценку
По каждой задаче определяются самые удачные её решения, отправленные участником. Технические баллы за них суммируются. Полученная сумма умножается на корректировочный коэффициент $\frac{10}{35}$ и округляется до целого. Результат определяет итоговую оценку.

Условия задачи 5

Цветочный город поразила эпидемия загадочного заболевания пурпурного пятого пальца. Доктору Пилюлькину удалось определить, что заболевание имеет вирусную природу и передается при непосредственном контакте мизинцами. Удалось также выявить «нулевого» пациента, который приехал зараженным из Зеленого города, и отследить все контакты, приведшие к заражениям.

Вирус болезни кодирует свою наследственную информацию в РНК длины оснований. Было установлено, что при каждом заражении происходила мутация ровно одного из оснований РНК, и было установлено положение мутировавшего основания в РНК. РНК вируса можно представить как строку длины , состоящую из 4 букв A, C, G, U.

Если бы мутаций не было, то у всех заболевших РНК вируса полностью совпадали, но каждая мутация приводит к тому, что неизменная часть РНК вируса становится все меньше и меньше. Напишите программу, которая для двух пациентов определит длину самого большого совпадающего фрагмента РНК вирусов, заразивших этих пациентов. Например, если РНК вируса первого пациента кодируется как $AAGCUUCAG$, а РНК вируса второго пациента кодируется как $ACGCUUAAG$, то у этих РНК есть три совпадающих фрагмента: A, $GCUU$ и AG. Самый длинный фрагмент имеет длину 4. Обратите внимание, что фрагменты в двух РНК должны находиться на тех же самых местах.

У доктора Пилюлькина компьютер, к сожалению, достаточно старый, поэтому ваша программа должна экономно использовать ресурсы. Обратите внимание на ограничения по памяти в этой задаче. Memory RSS limit: 128Mb Stack limit: 8Mb.

`Формат ввода:`

Первая строка входных данных содержит три целых положительных числа M, N, K. M – это длина РНК вируса ($M < 801$), N – количество пациентов ($N < 1000001$), K – количество запросов на определение длины ($K < 1001$). Следующие $N - 1$ строк содержат описания контактов, приведших к заражению. Каждый контакт – это три числа S, D, B, обозначающие, что пациент S заразил пациента D (пациенты нумеруются подряд от 1 до N включительно), при этом произошла мутация в основании B (основания нумеруются подряд от 0 до $M - 1$ включительно). Гарантируется, что контакты описывают одно дерево заражений, в котором «нулевой» пациент имеет номер 1. Следующие K строк содержат описания запросов в виде пар чисел P, Q, где P и Q — номера пациентов, для которых нужно определить гарантированную длину самого большого совпадающего фрагмента РНК. Поскольку не даны сами РНК вирусов, а только места мутаций, возможно, что совпадающие фрагменты могут быть и длиннее за счет мутаций, которые компенсируют друг друга. Это учитывать не нужно.

`Формат вывода:`

Для каждого запроса выведите одно число – длину самого большого совпадающего фрагмента РНК.

`Пример ввода:`

```
16 3 1
1 2 6
2 3 12
3 1
```

`Пример вывода:`

```
6
```