

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ
И ИНФОРМАТИКИ**

Кафедра информационных систем управления

С. И. Кашкевич

**СБОРНИК
ОЛИМПИАДНЫХ ЗАДАЧ
ПО ИНФОРМАТИКЕ**

**Практикум для студентов
факультета прикладной математики и информатики**

В пяти частях

Часть 3

**МИНСК
2022**

УДК 004(075.3)
ББК 32.81к2
К31

Рекомендовано советом
факультета прикладной математики и информатики
4 февраля 2022 г., протокол № 6

Рецензент
заведующий кафедрой дискретной математики и алгоритмики,
доктор физико-математических наук, профессор *В. М. Котов*

Кашкевич, С. И.
К31 Сборник олимпиадных задач по информатике : практикум.
В 5 ч. Ч. 3 / С. И. Кашкевич. – Минск : БГУ, 2022. – 52 с.

Практикум содержит условия, а также разборы решений заданий второго этапа республиканской олимпиады по учебному предмету «Информатика», проведенных в г. Минске в 2016 – 2021 годах

Предназначен для студентов факультета прикладной математики и информатики, а также для школьников, готовящихся к поступлению в вузы.

УДК 004(075.3)
ББК 32.81к2

© Кашкевич С. И., 2022
© БГУ, 2022

ПРЕДИСЛОВИЕ

Школьные олимпиады по информатике проводятся уже достаточно давно. Каждый год они дают возможность школьникам получить льготы при поступлении в высшие учебные заведения, а в будущем – и возможность получить престижную работу, так как все больше различных ИТ-компаний на собеседованиях предлагают решить различные алгоритмические задачи. Именно на таких задачах и сфокусированы данные соревнования, а вместе с ними и данное издание. Кроме этого, мы считаем, что решение олимпиадных задач по информатике позволяет развивать логическое мышление и находить оптимальные решения даже в самых повседневных ситуациях.

Автор предлагаемого вашему вниманию практикума в течение многих лет составляет задачи для школьных и студенческих олимпиад различного уровня. Естественно, возникло желание предложить часть из накопленного опыта на суд общественности, издав сборники олимпиадных задач с их разборами. В 2012 году началась работа над изданием таких сборников. Три части, названные «Задачи школьных олимпиад по информатике», вышли в 2013 - 2014 годах. В 2016 году вышла первая часть практикума «Сборник олимпиадных задач по информатике», а в 2018 – вторая.

Эти книги можно скачать из электронной библиотеки БГУ по следующим адресам:

<https://elib.bsu.by/handle/123456789/41655> («Задачи школьных олимпиад по информатике», часть I)

<https://elib.bsu.by/handle/123456789/48379> (часть II)

<https://elib.bsu.by/handle/123456789/104667> (часть III)

<https://elib.bsu.by/handle/123456789/147857> («Сборник олимпиадных задач по информатике», часть I)

<https://elib.bsu.by/handle/123456789/211379> (часть II)

В третьей части сборника собраны условия заданий второго этапа республиканских олимпиад по информатике, проведенных в 2016 – 2021 годах под эгидой факультета прикладной математики и информатики БГУ, а также показаны авторские подходы к решению этих задач. Однако мы советуем читателям не спешить читать разбор заданий. Постарайтесь сначала попробовать придумать математическую модель для задачи, где это возможно, найти алгоритм решения задачи, подумать над деталями реализации и даже реализовать свое решение на известном Вам языке программирования. Не забывайте обдумать различные крайние случаи для решения, так очень частыми ошибками являются упущенные из вида случаи граничных значений параметров.

Тесты к задачам находятся по адресу:

https://drive.google.com/file/d/1z7_KJ5Ztw_zpGv1L26O8ozPoYKTokG_F/view?usp=sharing

Практикум рекомендуется студентам факультета прикладной математики и информатики, школьникам, участвующим в олимпиадном движении по информатике или планирующим принимать такое участие, а также всем, кто интересуется олимпиадным движением.

Задачи для олимпиад различного уровня никогда не готовятся в одиночку – слишком велика вероятность ошибки в идее задачи, её условиях, тестах. В работе над олимпиадными задачами, описанными в этом издании, мне помогали А. А. Толстиков и (для последних двух олимпиад) Д. Н. Горбач.

Автор благодарит Н. С. Юрчика, который внимательно прочёл рукопись и сделал ряд замечаний и предложений (особенно касающихся разборов задач), улучшивших её качество.

Выражаю также благодарность доктору физико-математических наук, профессору В. М. Котову за рецензирование рукописи.

Замечания и предложения по усовершенствованию этой работы просим присылать на электронный адрес kash@bsu.by.

УСЛОВИЯ ЗАДАЧ

2016/17 учебный год

Полное решение каждой задачи оценивалось в 100 баллов.

Имена входных файлов для всех задач: input.txt

Имя выходных файлов для всех задач: output.txt

Задача 1: Юные программисты

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Пятиклассник Кирилл осваивает азы программирования под чутким руководством своего старшего брата Глеба. Сейчас Кирилл изучает работу оператора **for** ...

Глеб дал брату следующую задачу: ввести три целых положительных числа a , b и N , а затем вывести последовательность чисел a , $a+b$, $a+2 \cdot b$, ..., $a+N \cdot b$. Увы, Кирилл забыл вывести пробел в качестве разделителя между числами, и в результате получилась длинная строка, состоящая из цифр... Глеб, конечно, указал брату на его ошибку и заставил исправить программу, а сам задумался: как определить k -ю цифру в строке, выведенной первым вариантом программы (нумерация цифр в строке начинается с единицы)? К сожалению, Глеб не нашёл красивого решения этой задачи и предлагает её вам.

Формат входных данных. Единственная строка входного файла содержит числа a , b , N , k ($1 \leq a, b \leq 1000$, $1 \leq N, k \leq 10^9$). В 90 % тестов N не превосходит 10^6 , в 50 % тестов N не превосходит 10^3 , а k — 10^5 , а в 20 % тестов эти величины не превосходят соответственно 100 и 200.

Формат выходных данных. Выведите найденную k -ю цифру строки. Если длина строки меньше, чем k , выведите -1.

Примеры входных и выходных данных

3 5 6 9	2
2 7 2 10	-1

Задача 2: За верность игре

Ограничение по времени: 0,25 секунды

Ограничение по памяти: 16 мегабайт

В карточной игре «Преферанс», реализованной на портале minigames.mail.ru, есть любопытная система бонусов, названная «за верность игре». Суть её заключается в следующем: один раз в сутки во время входа в игру игрок получает бонус либо в фишках (которыми можно сделать ставку в игре), либо золотыми монетами (за которые приобретаются дополнительные услуги, например, возможность смухлевать в игре «Подкидной дурак»).

Размер бонуса вычисляется следующим образом: год делится на промежутки в 5 дней, и 1 января считается первым днём промежутка. Для игрока того уровня, к которому относится один из авторов задачи, размер бонусов считается следующим образом:

- в первый день — 1000 фишек;
- во второй день — 5000 фишек;
- в третий и четвёртый день — по 3000 фишек;
- в пятый день — 3 золотые монеты

Иван Ильич, страстный любитель преферанса, получающий бонусы за верность игре в описанном выше размере, решил подсчитать, сколько таких бонусов он получил в течение года. Увы, он скоро запутался и попросил участников олимпиады помочь ему...

Формат входных данных. Первая строка входного файла содержит признак високосного года (1 — год високосный, 0 — нет). Во второй строке записано число K — количество дней, в которые Иван Ильич играл на портале. Это число положительно и не превосходит количества дней в соответствующем году. Наконец, последующие K строк содержат записи о дате игры в формате «ДД/ММ». Никакой сортировки дат нет, как нет и повторяющихся записей.

Формат выходных данных. Выведите два числа: количество фишек и золотых монет, заработанных Иваном Ильичом за верность игре.

Примеры входных и выходных данных

1	7000 0
5	
01/01	
31/03	
10/02	
31/12	
06/08	

0	5000 6
5	
01/01	
31/03	
10/02	
31/12	
06/08	

Задача 3: Дроби

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

N дробей заданы своими числителями и знаменателями. Сколько дробей из этого списка различны?

Обратите внимание, что не все дроби являются правильными — у некоторых числитель больше знаменателя, а у некоторых числитель и знаменатель не являются взаимно простыми числами.

Формат входных данных. Первая строка входного файла содержит величину N ($1 \leq N \leq 10^5$, для 50 % тестов эта величина не превосходит 1000). Далее следуют N строк, в каждой из которых записаны два целых числа A и B ($-10^9 \leq A \leq 10^9, 1 \leq B \leq 10^9$, для 90 % тестов $-10^5 \leq A \leq 10^5, 1 \leq B \leq 10^5$) — числитель и знаменатель очередной дроби.

Формат выходных данных. Выведите ответ на задачу — количество различных дробей.

Пример входных и выходных данных

6	22
3 9	
-2 1	
17 51	
0 4	
9 9	
0 1	

Задача 4: Считаем слова

Ограничение по времени: 0,25 секунд

Ограничение по памяти: 64 мегабайта

Непустая строка состоит из слов, разделённых одной или несколькими запятыми (все остальные символы считаются частью слова). В начале и в конце строки запятые могут быть, а могут и отсутствовать.

Найдите количество слов в этой строке, длина которых равна заданной величине K .

Формат входных данных. В первой строке входного файла записана величина K . Далее следует сама анализируемая строка. Её длина не превосходит 100000 символов (в 50 % тестов длина строки не превосходит 30000 символов, а в 20 % тестов — 255 символов). В строке содержатся только символы с кодами от 32 до 126. Величина K положительна и не превосходит длины строки.

Формат выходных данных. Выведите единственное число — ответ на задачу.

Примеры входных и выходных данных

3 ,, , 3,, ,,, 2, 1+2, 1 2	2
1 ,	0

2017/18 учебный год

Полное решение каждой задачи оценивалось в 100 баллов.

Имена входных файлов для всех задач: input.txt

Имя выходных файлов для всех задач: output.txt

Задача 1: Наведение порядка

Ограничение по времени: 0.25 секунд

Ограничение по памяти: 64 мегабайт

Василий Иванович, известный шахматист и дед шестиклассника Саши, приглашён руководить шахматным кружком в школу, в которой учится его внук.

Придя в первый раз в комнату, где размещается инвентарь кружка, Василий Иванович ужаснулся царящему там беспорядку (и понял, почему предыдущий руководитель кружка был уволен с «волчьим билетом»). Даже шахматные фигуры были свалены в один ящик вместо того, чтобы быть разобранными по комплектам. Скорее всего, некоторые фигуры потерялись...

Что же, надо начинать наводить в комнате порядок! В первую очередь Василий Иванович решил разложить фигуры по комплектам и поручил это дело своему внуку, а Саша просит участников олимпиады определить, сколько комплектов он может собрать.

Напомним, что шахматный комплект состоит из 32 фигур — по шестнадцать белого и чёрного цвета. В комплект одного цвета входят: по одному королю (king) и ферзю (queen), по две ладьи (rook), слона (bishop) и коня (knight), а также восемь пешек (pawn). В дальнейшем будем обозначать фигуры их цветом и названием, разделённым единственным пробелом, например, white king, black pawn, и т.д.

Формат входных данных. Первая строка входного файла содержит величину N — общее количество фигур, найденных в ящике ($1 \leq N \leq 500000$, в 90 % тестов эта величина не превосходит 100000, в 70 % тестов — 10000, а в 50 % тестов — 1000). Далее следуют N строк, каждая из которых содержит описание одной фигуры. Эти строки никак не упорядочены.

Формат выходных данных. Выведите единственное число — ответ на задачу.

Пример входных и выходных данных

40	1
black knight	
white rook	
white rook	
black pawn	
black pawn	
white pawn	
black king	
white queen	
white bishop	
black pawn	
black bishop	
black rook	
white knight	
black pawn	
white pawn	
white pawn	
white pawn	
white pawn	
white pawn	
white pawn	
white pawn	
black pawn	
white knight	
white king	
black pawn	
black pawn	
black knight	
black rook	
black bishop	
white bishop	
black knight	
white rook	
white rook	
black pawn	
black pawn	
white pawn	
black king	
black queen	
white bishop	
black pawn	

Задача 2: Первый опыт шифрования

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

На факультативе по информатике учитель рассказал школьникам о перестановочных методах шифрования. Их суть состоит в том, что все символы исходного текста сохраняются и в зашифрованном, но их порядок меняется.

По окончании занятия учитель предложил детям придумать свои методы шифрования и пообещал дать лучший из них в качестве задачи на второй тур республиканской олимпиады по информатике. Он выполняет своё обещание...

Шифруемое сообщение представляет собой одну строку. Она разбивается на последовательность *фрагментов* — подстрок длины K , где K — ключ шифрования. Последний фрагмент сообщения может, очевидно, иметь меньшую длину. После этого в каждом фрагменте первый символ меняется с последним, второй — с предпоследним, и т.д.

Применение этого же алгоритма к зашифрованной строке приводит к её расшифровке.

Для реальной работы этот шифр слишком примитивен, а вот для олимпиадной задачи вполне сгодится...

А вот и задача: зашифруйте исходную строку предложенным методом.

Формат входных данных. Первая строка файла содержит величину K ($2 \leq K \leq 1000$). Во второй строке записано шифруемое сообщение. Его длина не превосходит 30000 символов (в 50 % тестов она не превосходит 255 символов). В сообщении присутствуют только символы с кодами от 32 до 127, пробелов в начале и конце строки нет

Формат выходных данных. Единственная строка должна содержать зашифрованное сообщение.

Примеры входных и выходных данных

7 This is a secret message	si sihTrces a ssem teega
3 Cat	taC

Задача 3: Максимальный среди минимальных

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 мегабайт

Девятикласснику Пете неинтересно изучать тему «Одномерные массивы» на уроках информатики: он давным-давно научился работать с массивами во время подготовки к олимпиадам. Однако учитель требует выполнять все задания, многие из которых кажутся Пете слишком простыми.

Однажды Петя попросил учителя дать ему задание посложнее. Учитель предложил ему решить следующую задачу: найти максимальный элемент массива из числа повторяющихся минимальное количество раз. Например, для массива 6, 4, 6, 6, 4, 2, 2 таким элементом будет 4 (он повторяется два раза, а элемент со значением 6 — три).

Хоть и не с первого раза, но Петя справился-таки с этой непростой задачей. А сможете ли Вы решить её?

Формат входных данных. Первая строка входного файла содержит величину N — число элементов массива ($1 \leq N \leq 10^5$, в 50 % тестов эта величина не превосходит 1000, а в 80 % тестов — 10000). В последующих строках записаны одно или несколько целых чисел, не превосходящих по модулю 10^6 — элементы массива.

Формат выходных данных. Выведите два числа: искомый максимальный элемент и количество его повторений.

Пример входных и выходных данных

7	4 2
6 4 6 6	
4 2 2	

Задача 4: Преобразование последовательности

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

Задана последовательность $A = (A_1, A_2, \dots, A_K)$ из K целых положительных чисел. На основании её получите последовательность $B = (B_1, B_2, \dots, B_K)$, где B_i — максимальное из чисел, полученных из A_i заменой в его двоичном представлении (без ведущих нулей) одной единицы на ноль, либо одного нуля на единичку.

Формат входных данных. В первой строке записано число K ($1 \leq K \leq 10^5$). Вторая строка состоит из K чисел, не превосходящих $2^{32} - 1$

— последовательность **A** (для 50 % тестов эти числа не превосходят $2^{16}-1$). Числа B_i также не должны превосходить этой величины.

Формат выходных данных. Выведите строку из K чисел — элементов последовательности **B**. Числа должны разделяться единственным пробелом, в начале и конце строки пробелы не допускаются.

Пример входных и выходных данных

3	58 6 192
50 7 128	

из электронной библиотеки БГУ

2018/19 учебный год

Полное решение каждой задачи оценивалось в 100 баллов.

Имена входных файлов для всех задач: input.txt

Имя выходных файлов для всех задач: output.txt

Задача 1: Хорошие пароли

Ограничение по времени: 1 секунда

Ограничение по памяти: 128 мегабайт

В распоряжении организаторов олимпиад по информатике и программированию, проводимых в Белорусском государственном университете, имеется программа, которая автоматически генерирует пароли для участников различных соревнований. Генерация паролей проводится по следующим правилам:

- длина пароля равняется восьми символам;
- в состав пароля входят прописные и строчные латинские буквы, а также цифры;
- для того, чтобы не спутать похожие по начертанию символы, буква О (прописная буква o) и цифра 0 (цифра ноль), а также буквы I (прописная буква i), l (строчная буква L) и цифра 1 (цифра один) не включаются в состав символов пароля.

Задано множество строк. Определите, сколько из них могут быть сгенерированы описываемой программой.

Формат входных данных. Первая строка входного файла содержит величину N — мощность исходного множества строк. Далее следуют N непустых строк с элементами этого множества.

Величина N положительна и не превосходит 1000, а сами строки не превосходят по длине 30000 символов. В 50 % тестов все строки исходного множества имеют длину в восемь символов. В состав строк исходного множества могут входить только символы с кодами от 32 до 127.

Все строки не начинаются и не заканчиваются пробелами.

Формат выходных данных. Выведите два числа N_1 и N_2 — количество строк, полученных с помощью программы генерации паролей, и количество оставшихся строк. Обратите внимание, что $N_1 + N_2 = N$.

Примеры входных и выходных данных

7 NSPLLCT2 aLpSCpFk haXh97EX 4cbr7Bkh 4YdnSXCn AT4uHF5a skXR8BKZ	7 0
5 good password password bad password 12345678 8gF4IkaE	1 4

Задача 2: Экспрессные маршруты - 2

Ограничение по времени: 1 секунда

Ограничение по памяти: 128 мегабайт

Между городом **A** и городом **B** проложена единственная дорога, на которой построено N остановочных пунктов. Обычный автобусный маршрут из **A** в **B** предусматривает остановки на каждом из оборудованных пунктов. Экспрессный маршрут пропускает некоторые (не менее одного) остановочных пунктов, но ни один экспрессный маршрут, во-первых, не пропускает более одного пункта подряд, и во-вторых, не останавливается более чем в трёх пунктах подряд.

Сколько различных экспрессных маршрутов можно организовать между городом **A** и городом **B**? Два маршрута считаются различными, если множества остановочных пунктов, которые они пропускают, различны.

Формат входных данных. Входные данные содержат значение числа N ($1 \leq N \leq 100$; для 50 % тестов $N \leq 50$).

Решения, дающие правильный ответ для $N \leq 10$, смогут набрать 15 баллов.

Формат выходных данных. Выведите единственное число — искомое количество экспрессных маршрутов.

Примеры входных и выходных данных

1	1
5	10

Замечание. Первая задача этой серии предлагалась на районных олимпиадах г. Минска в 2007 году.

Задача 3: Заполнить матрицу

Ограничение по времени: 1 секунда

Ограничение по памяти: 128 мегабайт

Матрица из M строк и N столбцов заполняется натуральными числами, начиная с единицы, по следующему принципу:

- первая строка слева направо;
- оставшиеся элементы первого столбца сверху вниз;
- оставшиеся элементы второй строки справа налево;
- оставшиеся элементы второго столбца снизу вверх.

После этого такие же действия выполняются для оставшейся части матрицы.

Так, матрица из 6 строк и 5 столбцов будет выглядеть следующим образом:

1	2	3	4	5
6	14	13	12	11
7	18	19	20	21
8	17	22	26	25
9	16	23	28	29
10	15	24	27	30

Найдите значение числа, стоящего в i -й строке и j -м столбце полученной матрицы. Нумерация строк и столбцов начинается с единицы.

Формат входных данных. Единственная строка входного файла содержит четыре числа: M, N, i, j ($1 \leq M, N \leq 10^9, 1 \leq i \leq M, 1 \leq j \leq N$). В 50 % тестов значения M и N не превосходят 10^5 , а в 20 % тестов — 10^2 .

Формат выходных данных. Выведите единственное число — ответ на задачу.

Примеры входных и выходных данных

6 5 2 5	11
6 5 5 2	16

Задача 4: Железо любит ласку, чистоту и смазку...

Ограничение по времени: 2 секунды

Ограничение по памяти: 128 мегабайт

Василий Иванович поехал навести порядок на даче, доставшейся ему по наследству, и взял с собой сына-восьмиклассника Ивана. Копаясь в дальнем углу сарая, Ваня обнаружил несколько обрывков собачьей цепи. Увидев, что сын нашёл себе хоть какое-то занятие, Василий Иванович поручил собрать из этих обрывков одну длинную цепь, чтобы привязать новую собаку (которую они когда-нибудь заведут) к конуре.

Все звенья в обрывках, кроме крайних, соединены с двумя другими. Новая цепь должна иметь такую же структуру.

Каждое звено в найденных обрывках снабжено карабинным замком, который позволяет легко отсоединить это звено от соседних (или наоборот, соединить несколько обрывков).



Увы, с течением времени все замки пришли в негодность, и для того, чтобы восстановить их работоспособность, придётся немало повозиться: промыть, очистить от ржавчины, смазать... Ване не очень нравится эта работа, но поручение надо выполнять! Поэтому Ваня решил поработать только над теми замками, которые необходимо будет раскрыть для сбора цепи. Помогите ему, рассчитав минимальное количество таких звеньев!

Формат входных данных. Первая строка входного файла содержит величину K — число найденных обрывков ($2 \leq K \leq 300000$, в 50 % тестов эта величина не превосходит 1000). Далее следуют K строк, в каждой из которых записано одно число p_i — количество звеньев в каждом обрывке ($1 \leq p_i \leq 10000$).

Формат выходных данных. Выведите одно число — количество замков, которые необходимо починить.

Примеры входных и выходных данных

3 4 1 5	1
5 4 6 5 7 3	3

Замечание. В первом примере достаточно починить замок на втором обрывке. Во втором примере надо обработать все звенья последнего обрывка.

2019/20 учебный год

Полное решение каждой задачи оценивалось в 100 баллов.

Ввод и вывод во всех задачах выполняется из стандартных потоков, а не из файлов.

Задача 1: За верность игре - 2

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

На втором этапе республиканской олимпиады по информатике в 2016 году была предложена следующая задача:

В карточной игре «Преферанс», реализованной на портале minigames.mail.ru, есть любопытная система бонусов, названная «за верность игре». Суть её заключается в следующем: один раз в сутки во время входа в игру игрок получает бонус либо в фишках (которыми можно сделать ставку в игре), либо золотыми монетами (за которые приобретаются дополнительные услуги, например, возможность смухлевать в игре «Подкидной дурак»).

Размер бонуса вычисляется следующим образом: год делится на промежутки в 5 дней, и 1 января считается первым днём промежутка. Для игрока того уровня, к которому относится один из авторов задачи, размер бонусов считается следующим образом:

- в первый день – 1000 фишек;
- во второй день – 5000 фишек;
- в третий и четвёртый день – по 3000 фишек;
- в пятый день – 3 золотые монеты.

Иван Ильич, страстный любитель преферанса, получающий бонусы за верность игре в описанном выше размере, решил подсчитать, сколько таких бонусов он получил в течение года. Увы, он скоро запутался и попросил участников олимпиады помочь ему...

Много воды утекло с тех времён, и условия выдачи бонусов изменились ...

Теперь за пропуск хотя бы одного дня отсчёт начинается с начала. Так, если игрок участвовал в играх 1, 2, 4, 5 и 6 января, он получает бонус в 15000 фишек (а не в 10000 фишек и три монеты, как ранее).

Решите задачу в новых условиях!

Формат входных данных. Первая строка входного файла содержит признак високосного года (1 – год високосный, 0 – нет). Во второй строке записано число K – количество дней, в которые Иван Ильич играл на портале. Это число положительно и не превосходит количества дней в соответствующем году. Наконец, последующие K строк содержат записи о дате игры в формате «дд/мм». Никакой сортировки дат нет, как нет и повторяющихся записей.

Для 20 % тестов все даты относятся к одному месяцу, а 30 % тестов соответствуют високосному году.

Формат выходных данных. Выведите два числа – количество фишек и золотых монет, полученных Иваном Ильичом за верность игре.

Примеры входных и выходных данных

0 5 06/01 02/01 04/01 01/01 05/01	15000 0
1 5 01/01 31/03 10/02 31/12 06/08	5000 0
0 11 01/02 02/02 03/02 04/02 06/02 07/02 08/02 09/02 10/02 11/02 12/02	30000 3

Задача 2: Казино

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Только что открывшееся казино не устаёт завлекать посетителей всё новыми и новыми играми.

Одна из таких игр состоит в следующем: для игрока генерируются два числа A и B , и его просят сформировать максимальное число, сумма десятичных цифр которого равна A , а число десятичных разрядов не превосходит B .

Если игрок не сумел назвать этого числа или его сформировать невозможно, он платит штраф, в противном случае – получает выигрыш, равный полученному числу.

Увы, при выходе из игорного зала его останавливают работники налоговой инспекции и предлагают оплатить налог на выигрыш. Размер этого налога равен минимальному числу, построенному по этим же правилам. При проигрыше игрока налог, естественно, не платится.

Предполагается, что Вы можете посчитать ответ всегда, когда это возможно. Сколько денег Вы вынесете из казино?

Формат входных данных. Введите два числа A и B ($1 \leq A \leq 100000, 1 \leq B \leq 10000$, в 70 % тестов $1 \leq A \leq 200, 1 \leq B \leq 18$, а в 30 % тестов $1 \leq A \leq 100, 1 \leq B \leq 9$).

Формат выходных данных. Выведите единственное число – ответ на задачу. Если игрок платит штраф, выведите -1.

Примеры входных и выходных данных

25 3	198
40 4	-1
1 4	999

Задача 3: Расширение строки

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Задана непустая строка A и символ b . Над строкой A выполняется следующее преобразование: она дублируется, а затем справа приписывается символ b . Затем такое же преобразование несколько раз выполняется над полученной строкой.

Так, если $A = 'abc'$, а $b = 'd'$, то в результате первых трёх преобразований получим строки

abcabcd

abcabcdabcabcd

abcabcdabcabcdabcabcdabcabcdab

Определите, какой символ будет стоять в N -й позиции строки слева (нумерация позиций начинается с единицы) после достаточного количества таких преобразований.

Формат входных данных. В первой строке записана строка A , во второй – символ b , в третьей – величина N ($1 \leq N \leq 10^{18}$, в 50 % тестов $N \leq 1000$, а в 20 % тестов $N \leq 100$). Длина строки A не превосходит 10, а в её состав могут входить только прописные и строчные латинские буквы, а также цифры. Это же относится к символу b .

Формат выходных данных. Выведите единственный символ – ответ на задачу.

Примеры входных и выходных данных

abc d 12	b
abc d 15	d

Задача 4: Две скважины

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Нефтедобывающая компания, в которой Вы работаете старшим помощником младшего аналитика, получила права на разработку нового нефтеносного участка. Этот участок имеет форму прямоугольника размером $M \times N$, разбитого для удобства разведки на квадраты единичной площади. Геологи оценили суточную добычу нефти с каждого квадрата в a_{ij} ($1 \leq i \leq M, 1 \leq j \leq N$).

Руководство компании решило для начала установить на участке две скважины и поручило Вам определить их положение.

Скважина занимает один из участков и выкачивает нефть с него в полном объеме суточной добычи. Кроме того, она может выкачивать половину суточной добычи с соседних (по вертикали, горизонтали и диагонали) участков.

Вы предложили следующий алгоритм решения этой задачи: первую скважину ставим на тот участок, на котором обеспечивается максимальный объем добычи. Вторую скважину ставим так, чтобы она не мешала первой (т. е. ни один участок не был бы соседним для обеих скважин) и обеспечивала бы максимальный объем добычи. Если существует несколько вариантов установки первой скважины, из них выбирается тот, при котором объем добычи второй максимален.

Несмотря на то, что предложенный алгоритм не является оптимальным, начальство утвердило его...

А теперь задача – напишите программу, рассчитывающую суммарную суточную добычу нефти с обеих скважин при применении указанного алгоритма.

Формат входных данных. В первой строке записаны величины M и N ($1 \leq M, N \leq 1000$, $6 \leq \max(M, N)$). Далее следуют M строк по N чисел в каждой – целочисленные величины a_{ij} ($1 \leq a_{ij} \leq 1000$). Для 30 % тестов $M, N \leq 100$, а для 70 % тестов $M, N \leq 500$.

Формат выходных данных. Выведите единственное число – ответ на задачу.

Примеры входных и выходных данных

3 6 10 11 3 5 8 9 8 9 3 7 5 4 3 4 2 6 3 7	60.5
1 6 10 5 9 3 4 7	23.5
1 6 2 10 10 10 10 1	26.0

Замечание. В первом примере скважины должны быть установлены на участках с координатами (2, 2) и (2, 5). Во втором примере первая скважина устанавливается на участке (1, 2), а вторая – либо на участке (1, 5), либо на (1, 6). В третьем примере первая скважина устанавливается на участке (1, 4), а вторая – на участке (1, 1).

2020/21 учебный год

Полное решение каждой задачи оценивалось в 100 баллов.

Ввод и вывод во всех задачах выполняется из стандартных потоков, а не из файлов.

Задача 1: Железнодорожная изгородь - 2

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мебибайт

На втором этапе республиканской олимпиады по информатике 2015 года была предложена следующая задача:

Метод «железнодорожной изгороди» (по правде говоря, авторы задачи не понимают смысл этого термина) — один из простейших методов шифрования текстовых сообщений, разработанный в начале XX века. Конечно, сейчас говорить о его надёжности и стойкости смешно, но, тем не менее, в анналы криптографии он занесен...

Суть метода «железнодорожной изгороди» заключается в следующем: текст длиной N символов посимвольно записывается «змейкой» в таблицу из N столбцов и k строк (здесь k — ключ шифрования). Пример заполнения таблицы показан на рисунке для $k = 5$ и исходной строки `This_is_a_secret_message`, для наглядности пробелы заменены символом нижнего подчёркивания.

T					a						-						
	h				-	-			t	m						e	
	i		s			s		e			e				g		
		s	i				e	r				s		a			
							c						s				

После заполнения таблицы символы исходного текста читаются из этой таблицы построчно. Таким образом, зашифрованная строка имеет вид Ta_h_tmeisseegsiersa_cs. Заметьте, что длина зашифрованного сообщения всегда совпадает с длиной исходного.

Вам требуется зашифровать сообщение методом «железнодорожной изгороди».

А вам предлагается решить обратную задачу – расшифровать текст, зашифрованный методом «железнодорожной изгороди».

Формат входных данных. Первая строка входного файла содержит величину ключа k ($2 \leq k \leq 1000$). Во второй строке записано зашифрованное сообщение, длина которого не меньше 1 символа и не превосходит 30000 символов (в 50 % тестов эта длина не превосходит 255). В сообщении могут входить символы с кодами от 32 до 127, пробелов в начале и конце строки нет.

Формат выходных данных. Выведите единственную строку с расшифрованным сообщением.

Примеры входных и выходных данных

5 Ta h tmeisseegsiersa cs	This is a secret message
8 Cat	Cat

Задача 2: Непокрывающиеся интервалы

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

На числовой оси расположены N интервалов, заданных своими начальными и конечными точками (a, b) . Будем говорить, что интервал (a_1, b_1) покрывается интервалом (a_2, b_2) , если $a_2 \leq a_1$ и $b_1 \leq b_2$.

Определите количество заданных интервалов, которые не покрываются ни одним из других.

Формат входных данных. В первой строке задана величина N – количество интервалов ($2 \leq N \leq 500000$, для 80 % тестов $N \leq 100000$, для 60 % тестов $N \leq 10000$, для 40 % тестов $N \leq 1000$). Последующие N строк содержат целочисленные границы каждого интервала (a, b) ($-10000 \leq a, b \leq 10000$, $a < b$).

Формат выходных данных. Выведите единственное число – ответ на задачу.

Примеры входных и выходных данных

3 -1 8 2 15 -3 10	2
----------------------------	---

3	1
-1 8	
-1 8	
-1 8	

Задача 3: «Чужие» банкоматы

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Как Вам, наверное, известно, банки Беларуси за снятие наличных в банкоматах «чужих» банков (не в собственных банкоматах и не в банкоматах банков-партнёров) вводят комиссию, размер которой определяется банком, выпустившим карту.

Вы находитесь в длительной командировке в местности, где все банкоматы относятся только к «чужим» банкам. Банк, в котором выпущена Ваша карточка, установил следующий размер комиссии: p % от суммы снимаемых средств, но не менее q рублей за операцию.

Но жить в чужом городе без наличных нельзя, и время от времени Вам приходится идти в «чужие» банкоматы...

Сколько денег (в рублях) спишется с Вашего счёта в результате выполнения всех операций по снятию наличных?

Обратите внимание, что сумма наличных для снятия кратна 5 рублям, однако учёт денег на счёте ведётся с точностью до копейки! Округление списываемых сумм выполняется по правилам округления вниз, т. е. 10.8 копеек округляются до 10.

Формат входных данных. В первой строке записаны величины p и q (неотрицательные, заданные с точностью до двух знаков в дробной части, $p \leq 5$, $q \leq 10$). Во второй строке задана величина N – количество выполненных операций по снятию наличных ($1 \leq N \leq 10^3$). Наконец, в третьей строке записаны N положительных целых чисел, кратных 5 – сумма наличных, снимаемых при каждой операции. Одна операция позволяет снять не более 1000 рублей.

Дополнительные ограничения:

- в 50 % тестов значение p и q целые, суммы наличных кратны 100;
- в 80 % тестов значение p и q целые.

Формат выходных данных. Выведите единственное число – сумму денег, списанных с Вашего счёта. Эта сумма должна быть выведена с точностью до 0.01.

Примеры входных и выходных данных

4 10 5 100 200 300 400 500	1568.00
3 10 10 100 100 200 205 300 350 410 450 515 50	2791.75
3 1.20 7 100 50 5 70 300 45 25	614.35

Задача 4: Подматрицы

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

Пусть задана целочисленная матрица A из M строк и N столбцов. Назовём *подматрицей* этой матрицы прямоугольную часть исходной матрицы, начиная с ячейки (i_1, j_1) и заканчивая ячейкой (i_2, j_2) .

Вам требуется выполнить Q запросов на нахождение суммы элементов в различных подматрицах матрицы A .

Формат входных данных. Первая строка содержит величины M , N , Q ($1 \leq M, N, Q \leq 100000$, $N \times M \leq 1000000$). Далее следуют M строк из N чисел каждая – элементы матрицы A . Модули значений элементов не превосходят 10^6 .

После этого следуют Q строк, каждая из которых содержит параметры одного запроса – величины i_1, j_1, i_2, j_2 ($1 \leq i_1 \leq i_2 \leq M$, $1 \leq j_1 \leq j_2 \leq N$).

Дополнительные ограничения:

- в 20 % тестов $2 \leq M, N, Q \leq 100$;
- в 30 % тестов $M = 1$.

Формат выходных данных. Выведите в одну строку Q чисел – результаты выполнения запросов. Порядок следования этих чисел должен соответствовать порядку запросов во входном файле.

Пример входных и выходных данных

5 6 4	115 36 48 15
-3 6 9 0 5 2	
11 4 3 8 9 0	
4 7 2 1 8 7	
9 8 3 2 1 0	
8 -3 2 -5 6 1	
1 1 5 6	
2 4 4 6	
2 3 5 6	
1 2 1 4	

из электронной библиотеки БГУ

2021/22 учебный год

Полное решение каждой задачи оценивалось в 100 баллов.

Ввод и вывод во всех задачах выполняется из стандартных потоков, а не из файлов.

Задача 1: Левая рука тьмы

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

В романе «Левая рука тьмы» американской писательницы Урсулы Ле Гуин, удостоенном обеих высших наград американской фантастики: Небьюла (1969) и Хьюго (1970), описана удивительная картина мира планеты Зима. Жители этой планеты являются гермафродитами. Большую часть времени они бесполоы. Лишь в течение периода половой активности, длящегося 2-5 дней в месяц, они обзаводятся половыми признаками и могут стать либо мужчиной, либо женщиной (но не мужчиной и женщиной одновременно). Зачатие возможно только во время половой активности обоих партнёров (мужчины и женщины).

Таким образом, один и тот же индивид может быть отцом одних детей и матерью других.

А теперь – задача:

Записи актов о рождении содержат три поля: имя отца, имя матери и имя ребёнка. На основании этой информации определите, для скольких детей каждый из упомянутых жителей является матерью, а для скольких – отцом.

Предполагается, что все имена уникальны. Тесты подобраны таким образом, что никто не является предком или потомком самого себя (однако допускается ситуация, когда один и тот же индивид родил сына, будучи мужчиной, а через некоторое время, став женщиной, родил от своего сына или одного из его потомков другого ребёнка).

Формат входных данных. Первая строка содержит N – количество записей актов о рождении ($1 \leq N \leq 10000$, для 30 % тестов $N \leq 100$, а для 50 % тестов $N \leq 500$). Далее следуют N строк с одной записью о рождении каждая. Эти строки имеют следующий формат:

имя отца, имя матери, имя ребёнка

Длина имени не превосходит 100 символов, в его состав входят символы с кодами от 32 до 127 (включая пробелы), кроме запятой. Имена не начинаются и не заканчиваются пробелом.

Никакой отсортированности записей о рождении нет.

Формат выходных данных. Выведите несколько записей, количество которых соответствует количеству упомянутых во входных данных жителей планеты Зима.

Каждая запись имеет формат:

имя, t_1, t_2

где t_1 и t_2 – количество жителей, для которых указанный индивид был отцом и матерью соответственно.

Выходные строки должны быть отсортированы по возрастанию имён.

Пример входных и выходных данных

4	Person 1, 1, 0
Person 6, Person 2, Person 5	Person 2, 1, 1
Person 4, Person 5, Person 1	Person 3, 0, 0
Person 2, Person 6, Person 4	Person 4, 1, 0
Person 1, Person 6, Person 3	Person 5, 0, 1
	Person 6, 1, 2

Задача 2: Галлеоны, сикли и кнаты

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Оказывается, маги тоже пользуются деньгами! У них в ходу золотые, серебряные и бронзовые монеты, называемые соответственно галлеонами, сиклями и кнатами. Один галлеон содержит семнадцать сиклей, а один сикль – двадцать девять кнатов. Все денежные данные (цены, наличные деньги) формулируются в галлеонах, сиклях и кнатах, причём количество сиклей не превышает 16, а количество кнатов – 28.

Гарри Поттер перед поступлением в Хогвардс получил в банке «Гринготс» на обзаведение определённую сумму. Однако поступающим надо купить абсолютно необходимые для учёбы вещи: волшебную палочку, сову, мантии, оловянный котёл, медные весы...

Сколько денег окажется у Гарри Поттера после всех покупок?

Формат входных данных. Первая строка содержит количество денег, полученных Гарри в банке «Гринготс» (три числа – количество галлеонов, сиклей и кнатов, записанные через пробелы. Напомним, что количество сиклей не превышает 16, а количество кнатов – 28). Во второй строке записано число N – количество предметов, которые необходимо

купить. Далее следуют N строк с ценами каждого предмета (в том же формате, что и первая строка). Каждый купленный предмет стоит не менее одного кнута.

Ограничения на данные: $0 \leq N \leq 10^6$ (в 40 % тестов $N \leq 10^5$), количество галлеонов в каждой из строк также не превосходит 10^6 .

Формат выходных данных. Единственная строка содержит информацию о количестве денег, оставшихся у Гарри после совершения всех покупок (в том же формате, что и первая строка входного файла). Если денег не хватит, выведите в выходной файл число -1.

Примеры входных и выходных данных

5 16 10 2 3 9 21 0 4 0	2 2 18
1000 0 1 5 200 0 1 200 0 1 200 0 1 200 0 1 200 0 1	-1

Задача 3: Дребезжащие клавиши

Ограничение по времени: 1 секунда

Ограничение по памяти: 256 мегабайт

Моя клавиатура служит мне верой и правдой уже пятнадцать с лишним лет. Увы, всякой вещи приходит пора уходить на покой... Я заметил, что нажатие на отдельные клавиши приводит к повторению символа, соответствующего нажатой клавише (эта проблема называется «дребезгом клавиш»). Например, вместо слова **"abc"** я могу получить **"abbbbcccc"**.

А теперь – задача:

Введены две строки **A**, которую я хочу набрать, и **B**. Определите, может ли строка **B** быть результатом дребезга клавиш, либо я в процессе ввода допустил опisku. Имейте в виду, что дребезга при вводе строки может и не быть!

Формат входных данных. Первая строка содержит величину N ($2 \leq N \leq 100$) – количество подтестов. Далее записаны N пар непустых строк, которые интерпретируются как **A** и **B** для очередного подтеста. Длина строк не превосходит 30000 символов. Для 50 % тестов $N \leq 50$, а длины строк не превосходят 255 символов. Все коды символов в строках находятся в интервале от 32 до 126.

Формат выходных данных. Выведите строку из N символов. Каждый символ соответствует очередному подтесту и равен 'Y', если вторую строку можно объяснить дребезгом клавиш, или 'N' в противном случае.

Пример входных и выходных данных

5 This is a simple text This is a simple text This is a simple text this is a ssimple texxt 12345 111122333444555 abc aabbabbbbbbcb abs aabbabbbbbbcb	YNYYN
---	-------

Задача 4: Закупка урожая

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

План поля представляет собой прямоугольник размером $M \times N$, разбитый на квадратные участки единичной площади, причём границы участков параллельны границам поля. На участке с координатами r , с построен склад зерна ёмкостью Q тонн, остальные участки сданы в аренду различным арендаторам, которые засеяли свою землю одной и той же культурой.

Наступило время сбора урожая... Каждый арендатор определяет цену, за которую он готов продать часть урожая на склад, и количество продаваемого урожая. Так, с участка с координатами i, j продаётся b_{ij} тонн зерна по цене a_{ij} у.е. за тонну. Владелец склада дополнительно оплачивает расходы по доставке с этого участка, которые составляют $T * (|r - i| + |c - j|)$ у.е. за тонну.

Естественно, владелец желает заполнить склад зерна, затратив как можно меньше денег... Ему посоветовали вначале закупать зерно с тех

участков, для которых суммарная стоимость закупки и доставки одной тонны зерна минимальна. (Естественно, если предлагаемый к закупке объём зерна со всех участков не превосходит Q , следует закупить всё предложенное зерно, а потом искать другие пути пополнения склада) Несмотря на возражения, что этот механизм не является оптимальным, Вам, как наёмному аналитику, поручено рассчитать сумму, которую необходимо заплатить за зерно нового урожая. Справитесь?

Формат входных данных. В первой строке записаны величины M, N, r, c, Q, T . Далее следуют M строк из N чисел каждая – величины b_{ij} . Наконец, записываются M строк из N чисел каждая – величины a_{ij} .

Ограничения на данные: $1 \leq M, N \leq 1000, M * N > 1, 1 \leq Q \leq 100000, 0 \leq T \leq 100000, 1 \leq r \leq M, 1 \leq c \leq N, 0 \leq a_{ij}, b_{ij} \leq 1000$.

Гарантируется, что $a_{rc} = b_{rc} = 0$.

Формат выходных данных. Выведите единственное число – ответ на задачу.

Пример входных и выходных данных

5 4 2 2 100 8	1 6 6 1
10 15 21 43	
19 0 21 45	
10 6 17 24	
30 16 19 2	
30 70 40 5	
14 6 10 9	
9 0 8 2	
6 7 3 2	
6 7 9 1	
8 6 4 1	

РАЗБОР ЗАДАЧ

2016/17 учебный год

Задача 1: Юные программисты

Самая сложная задача соревнования. Неспроста полное решение на олимпиаде выполнил лишь один участник...

Рассмотрим два подхода к решению этой задачи. Первый, к сожалению, не укладывается в отведенное время, а вот второй работает правильно и набирает 100 баллов.

- Будем рассчитывать каждое из чисел a , $a+b$, $a+2 \cdot b$, ..., $a+N \cdot b$, определять текущую длину для десятичного представления этого числа и накапливать общую длину последовательности. Как только она превысит K , прекращаем расчёты – искомая цифра находится среди цифр последнего члена последовательности.

Не забываем, что для хранения последних элементов 32-битового типа данных не хватит, необходимо использовать типы `int64` (в Паскале) или `long long` в C++.

Как уже было сказано ранее, авторское решение, основанное на этом подходе, набирает 92 балла из 100 – при предельных значениях параметров оно не укладывается в лимит по времени.

- В зависимости от входных данных определяем величины t_1 , t_2 , ..., t_p , ... - количество однозначных, двузначных, и т.д. элементов последовательности (вывод соответствующих формул предлагается читателю). Определяем финальное значение P как минимальное из двух величин

$$\sum_{i=1}^P t_i \leq N$$

и

$$\sum_{i=1}^P i * t_i \leq k$$

После этого высчитываем конечный элемент последовательности и определяем нужную цифру.

Задача 2: За верность игре

А эта задача – самая простая...

Заводим массив целых чисел – номер первого дня месяца относительно начала года:

days: array [1..12] of integer = (0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334);

После этого для каждой даты, когда Иван Ильич заходил на игровой портал, рассчитываем номер этого дня относительно начала года (не забывая сделать поправку на високосный год), и по остатку от деления этой величины на 5 определяем полагающийся игроку бонус.

Задача 3: Дроби

В задаче есть подводный камень, на который, наверное, натолкнулись многие участники...

Нельзя приводить дробь к типу с дробными частями! Дело в том, что при таком преобразовании дроби, отличающиеся на одну миллиардную (а это допустимо по условию задачи) будут считаться одинаковыми!

Правильное решение заключается в выполнении следующих шагов:

- приводим очередную дробь к каноническому виду (делаем знаменатель положительным, в случае нулевого числителя приравниваем знаменатель к единице, сокращаем дробь при необходимости);
- помещаем полученную пару «числитель – знаменатель» в контейнер, не допускающий хранения повторяющихся значений (для языка C++ идеальным, по мнению автора, будет `set <pair <int, int>>`);
- после обработки всех данных определяем число элементов, хранящихся в контейнере.

Трудоёмкость такого алгоритма – $O(N \log N)$, что является достаточным для полного решения задачи.

Задача 4: Считаем слова

Это – типичная задача на обработку строк. Автор постоянно даёт подобного рода задачи первокурсникам при изучении соответствующей темы...

Определим условие, когда символ находится в начале слова: он – не запятая, и он либо стоит в начале строки, либо перед ним стоит запятая. Обратите внимание, что менять порядок проверки последних двух условий нельзя!

Аналогичным образом описываем условие, когда текущий символ стоит в конце слова.

Теперь при определении начала слова запоминаем позицию соответствующего символа, а при нахождении конца – определяем длину слова и при необходимости увеличиваем счётчик результатов.

Задача 1: Наведение порядка

Жёсткое ограничение на время выполнения должно навести на мысль о том, что сортировка входных данных не нужна... Действительно, достаточно завести массив целых чисел из 12 элементов (по числу различных фигур), и в процессе ввода исходных данных накапливать в нём количество соответствующих фигур.

После этого следует найти частные от деления количества накопленных фигур на известное число фигур в комплекте. Минимальная из полученных величин и будет ответом на задачу.

Большой объём обрабатываемых данных требует аккуратного ввода (иначе окажется, что ввод данных требует больше времени, чем отведено на решение задачи). В частности, использование `getline` вместо `scanf` в авторском решении привело к нарушению лимита времени на 3 тестах из 50...

Задача 2: Первый опыт шифрования

Самая простая, по мнению авторов, задача в соревновании. Она является усложнением задачи «Перевернуть строку», предложенной на таких же соревнованиях в 2010/2011 учебном году (необходимо выполнить не один, а несколько переворотов). Всё, что нужно уметь для решения этой задачи – организовывать совместную работу с двумя индексами.

Определяем два целочисленных индекса i и j . Изначально $i = a, j = b$, где a и b – границы очередного фрагмента. Основной цикл обработки фрагмента S выглядит следующим образом:

```
while i < j do begin
    tmp := S[i];
    S[i] := S[j];
    S[j] := tmp;
    i := i + 1;
    j := j - 1;
end;
```

Стоит добавить только, что более эффективным методом для школьников, работающих на C++, на наш взгляд, будет использование алгоритма `reverse` из библиотеки STL.

Задача 3: Максимальный среди минимальных

Стандартная задача на обработку одномерных массивов. Похожие задачи автор даёт студентам-первокурсникам в качестве лабораторной работы повышенной сложности. Конечно же, для основной массы школьников эта задача слишком сложна, но на то и олимпиада...

При решении задачи можно использовать два подхода:

- накапливать в отдельном массиве количество повторений элементов исходного массива, а по окончании ввода просматривать полученный массив в поисках чисел, встречающихся минимальное число раз. Трудоёмкость этого подхода – $O(N^2)$, и авторское решение, реализующее такой алгоритм, нарушает предел времени на 5 тестах из 50;
- отсортировать исходный массив по возрастанию, а затем отслеживать изменения значений в отсортированном массиве в поисках ответа. Выбор оптимального алгоритма сортировки сводит трудоёмкость такого подхода к $O(N \log N)$, что является достаточным для полного решения задачи.

Ну и, естественно, не стоит забывать о потерях времени при использовании ввода через потоки C++. Но об этом уже говорилось в разборе задачи «Наведение порядка» ...

Задача 4: Преобразование последовательности

Для каждого элемента A_i исходной последовательности необходимо рассмотреть два варианта:

- $A_i = 2^p - 1$ для некоторого положительного p . В этом случае двоичное представление этого числа состоит только из единиц, и нам необходимо уменьшить A_i на единицу;
- в противном случае заменяем на единичку самый правый ноль, стоящий после ведущей единицы.

Для нахождения такой позиции можно воспользоваться как побитовыми операциями `and` и `or` (что предпочтительнее, но не все школьники владеют такими навыками), так и операциями деления на два и нахождения остатка. В последнем случае, однако, при использовании системы программирования Pascal ABC.Net приходится использовать нетривиальный тип `uint64` для хранения и обработки данных.

Задача 1: Хорошие пароли

Самая простая задача соревнования – её сделал 41 участник, в то время как каждую из оставшихся задач – не более 20 школьников.

Составляем строку **P** из символов, допустимых для паролей. После этого проверяем каждый из введенных паролей на соответствие двум критериям:

- 1) длина пароля равна 8 символам;
- 2) каждый из символов пароля входит в строку **P**.

Если оба критерия выполняются, увеличиваем N_1 , в противном случае – N_2 .

Задача 2: Экспрессные маршруты - 2

Задача требует знания основ динамического программирования. Естественно, она оказалась самой сложной в наборе...

Обозначим через $f(k)$ число экспрессных маршрутов в случае, если между конечными станциями построено k остановочных пунктов. Ответом на задачу, очевидно, будет вычисленная величина $f(N)$.

Для начальных значений k посчитаем величину $f(k)$ вручную. Этот расчёт сведён в соответствующую таблицу, где в описании маршрута ноль обозначает, что маршрут пропускает соответствующий пункт, а единица – маршрут останавливается на нём.

k	$f(k)$	маршруты
1	1	0
2	2	01, 10
3	4	010, 011, 101, 110
4	7	0111, 0101, 0110, 1011, 1010, 1101, 1110
5	10	01110, 01010, 01011, 01101, 10111, 10101, 10110, 11010, 11011, 11101
6	15	011101, 011011, 011010, 010110, 010111, 010101, 101110, 101101, 101011, 101010, 110111, 110101, 110110, 111011, 111010

Заметим, что условие «экспрессный маршрут останавливается не более чем на трёх остановочных пунктах подряд» действует, начиная с $k = 5$. Именно поэтому таблица начальных значений столь длинна...

Начинаем выводить рекуррентное соотношение. Возможны следующие случаи:

- на $k-1$ пункте автобус останавливается, а на k – нет;
- на k пункте автобус останавливается, а на $k-1$ – нет;

- автобус останавливается и на $k-1$, и на k пунктах.

Одному из первых условий удовлетворяют $f(k-1)$ маршрутов. Для выполнения третьего условия надо, чтобы описания маршрутов заканчивались на 1011 или на 0111. Таких маршрутов $f(k-3)$.

Итак, рекуррентное соотношение для вычисления $f(k)$ имеет вид

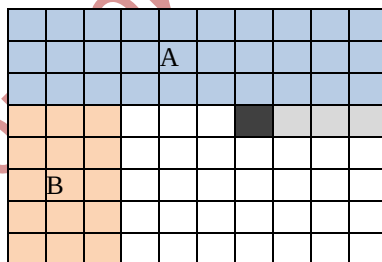
$$f(k) = f(k-1) + f(k-3), \text{ где } k \geq 7$$

Задача 3: Заполнить матрицу

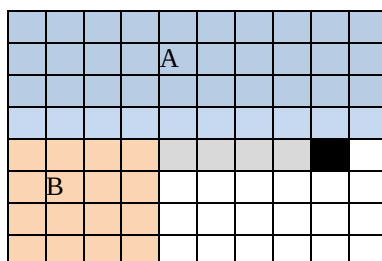
Первое, что может прийти в голову – явно строить матрицу до тех пор, пока не будет вычислен элемент в i -й строке и j -м столбце. Однако это, во-первых, сама по себе нетривиальная задача, а во-вторых, жёсткое ограничение по времени даст возможность пройти лишь порядка 50 % тестов.

Полное решение задачи потребует рассмотрения четырёх случаев.

1) $i \leq j, i$ – чётное. В этом случае заполнены два прямоугольника, обозначенные на рисунке буквами **A** и **B**, а также клетки i -й строки, расположенные правее искомой. Площадь прямоугольника **A** равна $(i-1) \times N$, площадь прямоугольника **B** – $(i-1) \times (N-i+1)$, а количество заполненных клеток последней строки (включая искомую) – $N-j+1$. Результат будет равен сумме этих выражений.

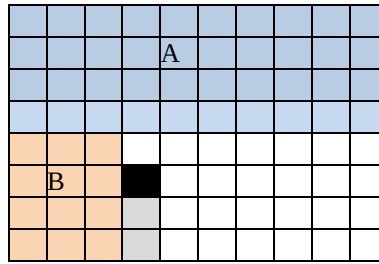


2) $i \leq j, i$ – нечётное¹

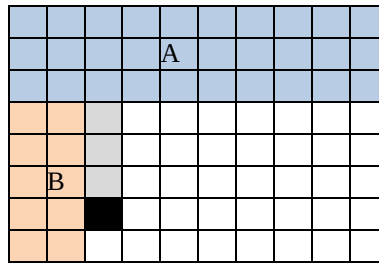


¹ В этом и последующих случаях автор лишь демонстрирует соответствующие рисунки, предоставляя читателям возможность рассчитать соответствующие выражения.

3) $i > j, j$ – чётное



4) $i \leq j, i$ – нечётное



Трудоёмкость решения этой задачи – $O(1)$.

Задача 4: Железо любит ласку, чистоту и смазку...

Задача «на сообразительность», не требующая никаких дополнительных знаний по информатике.

Отсортируем массив обрывков цепочек по возрастанию их длины. После этого начинаем обрабатывать самые короткие обрывки.

Пусть s – длина самого короткого обрывка, а M – количество оставшихся обрывков. Если $M - 1 \geq s$, чиним все звенья этого обрывка и соединяем ими $s + 1$ самых длинных обрывков и продолжаем эти же действия со следующим самым коротким обрывком, уменьшая величину M на $s + 1$. Если M станет равной единице, прекращаем работу.

В случае, когда $M - 1 < s$, нам достаточно починить $M - 1$ звено и соединить все обрывки в одну цепь. Работа опять-таки завершается.

Задача 1: За верность игре – 2

Самая простая задача соревнования – 55 полных решений...

В зависимости от того, високосный год или нет, строим булевский массив из 366 или 365 элементов. Каждый элемент этого массива равен `true`, если Иван Ильич играл в этот день, и `false` в противном случае.

Заполняем массив на основе анализа входных данных. После этого просматриваем его и для каждого дня рассчитываем его награду, в зависимости от длины непрерывного промежутка, когда Иван Ильич играл.

Ну, и пояснение того ограничения, что для 20 % тестов все даты относятся к одному месяцу. Оказывается, многие участники не знают, сколько дней в каждом месяце. Количество вопросов типа «Сколько дней в январе?» зашкаливало. Вот для таких школьников это ограничение и придумано...

Задача 2: Казино

Следующая по сложности задача (26 полных решений).

Прежде всего определяем, когда игрок платит штраф. Искомое число нельзя построить, если $9 \times B > A$. Если же $9 \times B = A$, единственное число, которое можно построить состоит из всех девяток. Размер налога равен размеру выигрыша, и игрок уходит из казино с пустыми руками.

Эти случаи рекомендуется рассмотреть до поиска общего решения.

Рассмотрим механизм образования максимального и минимального числа по описанным в условии задачи правилам. В максимальном числе значения старших разрядов полагаем равными 9, насколько это возможно. Далее записывается оставшаяся цифра, а младшие разряды заполняются нулями. Минимальное число (налог на выигрыш) формируется таким же образом, но движемся не от старших разрядов к младшим, а наоборот.

Так, для $A = 40$ и $B = 6$ максимальное и минимальное число равны соответственно 999940 и 49999 (незначащие нули в налоге можно отбросить).

Следующий этап – вычитание. И здесь нас ждёт ещё один подводный камень. Значение B слишком велико, чтобы использовать для хранения полученных чисел какой-то целочисленный тип.² В этом случае необходимо использовать т.н «длинную арифметику» - хранить число в виде массива цифр и моделировать вычитание «столбиком».

² Для тех участников, которые работают на языке Python, проблем не возникает – тип `int` имеет неограниченную размерность.

Сниженные ограничения позволяют работать и с целочисленными типами (70 % тестов – с 64-разрядными, 30 % тестов – с 32-разрядными).

Задача 3: Расширение строки

По мнению авторов, это – самая интересная задача соревнования...

Очевидно, прямое построение всех преобразований из-за больших ограничений на N даст возможность взять лишь 50 % тестов.

Авторское решение этой задачи предполагает использование механизма, схожего с дихотомией (хотя участники, особенно в процессе дорешивания, предложили много других красивых вариантов).

Прежде всего определим, сколько преобразований потребуется, чтобы длина строки стала не меньшей, чем N . Пусть f_k – длина строки после k преобразований. Посчитать эту величину можно исходя из следующего рекуррентного соотношения:

$$\begin{aligned}f_0 &= \text{length}(A); \\ f_k &= 2 \times f_{k-1} + 1;\end{aligned}$$

Последовательно производя расчёты по этому соотношению, добиваемся того, что $f_k \geq N$.

В рекурсивную подпрограмму, решающую эту задачу, будем передавать два целочисленных параметра `left` и `right` – границы обрабатываемого интервала. При первом вызове `left = 1`, `right =  $f_k$` .

Прежде всего проверяем условие $N = \text{right}$. Если оно выполняется, искомым символ равен b .

Теперь находим середину отрезка `[left, right]` и помещаем эту величину в переменную `middle := (left + right) div 2`. Поскольку длина отрезка `[left, right]` всегда нечётная, значение `middle` соответствует началу второго вхождения строки **A**.

Определяем, в каком из вхождений строки **A** находится N -й символ преобразованной строки:

```
if N < middle then
  right := middle - 1
else begin
  left := middle;
  right := right - 1;
end;
```

Следующая проверка: если $N - \text{left} + 1 \leq \text{length}(A)$, то искомым символ содержится в строке **A** по такому же индексу.

Если обе проверки не позволили определить искомым символ, запускаем подпрограмму рекурсивно.

Задача 4: Две скважины

Эта задача является усложнением задачи «Место для фабрики», предложенная на районных олимпиадах г. Минска в 2013/2014 учебном году. Её решение не требует знания каких-то специфических алгоритмов. Единственное, что требуется уметь – аккуратно и правильно работать с двумерными массивами (хотя работа с массивами, а особенно с двумерными, в настоящее время удалена из школьной программы).

Решаем задачу в два этапа. В начале построим матрицу **B** размерности $M \times N$, элементы которой равны количеству нефти, добываемой со скважины, находящейся в клетке (i, j) . Эта операция, несмотря на четыре вложенных цикла, имеет трудоёмкость $O(M \cdot N)$, поскольку внутренние циклы повторяются не более 9 раз.

Похожие циклы были приведены в предыдущих сборниках, но я обязан повторить его ещё раз:

```
for i := 1 to M do
  for j := 1 to N do begin
    for ki := -1 to 1 do
      for kj := -1 to 1 do begin
        if (i+ki < 1) or (i+ki > M)
           or (j+kj < 1) or (j+kj > N) then
          continue;
        if (ki=0) and (kj=0) then
          t := 1
        else
          t := 0.5;
        b[i, j] := b[i, j] + t * a[i+ki, j+kj];
      end;
    end;
  end;
```

На втором этапе находим пары (i, j) и (p, q) такие, что

- $b_{i,j} + b_{p,q}$ максимальна для всех таких пар;
- скважины, расположенные в этих точках, не мешают друг другу, т. е. $(|i - p| \geq 3) \text{ or } (|j - q| \geq 3)$.

Второй этап алгоритма имеет трудоёмкость $O(N^2 * M^2)$, что укладывается в имеющиеся ограничения (конечно, при наличии предпросчёта для вычисления матрицы **B**).

Задача 1: Железнодорожная изгородь – 2

Задача, оказавшаяся самой простой на соревновании (полное решение выполнили 46 участников основного соревнования и 54 участника зеркала).

Суть решения заключается в следующем: формируем массив **V** из k подстрок, соответствующих строкам таблицы шифрования, разбиваем исходную строку на подстроки, заполняя массив **V**. После этого результат получается чтением «змейкой» символов из строк этого массива, как сказано в описании алгоритма шифрования.

Обратите внимание, что элементы массива **V** имеют различную длину. Так, для первого примера из условия эти строки будут выглядеть следующим образом:

```
Ta_  
h__tme  
issee  
siersa  
_cs
```

Задача 2: Непокрывающиеся интервалы

Сортируем интервалы по возрастанию их начальных точек, а интервалы с одинаковыми значениями начальных точек – по убыванию конечных. После этой сортировки ни один из интервалов не будет покрываться интервалами, лежащими ниже. Первый из интервалов, очевидно, не покрывается ни одним из остальных, поэтому счётчик результата устанавливаем в единицу.

После этого задаём значение $t = b_1$ и последовательно просматриваем остальные интервалы. Если для какого-то из них $b_i > t$, то этот интервал не покрывается ни одним из предыдущих; полагаем $t = b_1$ и увеличиваем счётчик результата на единицу.

Следует сказать, что сортировка с трудоёмкостью $O(N^2)$ является слишком медленной (авторское решение, основанное на этом принципе, взяло 32 теста из 50). Для полного решения задачи требуется применять сортировку с трудоёмкостью $O(N \log N)$.

Особо следует сказать о втором примере, по которому было много вопросов во время олимпиады. Три одинаковых интервала – как их считать? В данном случае предполагается, что первый из интервалов (после сортировки) покрывает остальные, но не покрывается ими.

Задача 3: «Чужие» банкоматы

Авторы считали эту задачу самой простой в наборе (не надо знать никаких дополнительных алгоритмов, выходящих за рамки школьной программы, достаточно лишь выполнить грамотную реализацию). Однако она неожиданно получилась самой сложной (на основном турнире только один участник набрал полный балл, а на зеркале, где могли принять участие школьники любого региона Беларуси, полные решения выполнили лишь три человека).

Прежде всего следует сказать, почему было выбрано округление вниз. Стандартная функция `round` использует т.н. *бухгалтерское* округление, которое неизвестно подавляющему большинству школьников. Согласно правилам этого округления (при округлении до целого) в случае, если дробная часть равна 0.5, получившееся чётное число округляется в меньшую сторону, а нечётное – в большую. Иными словами, результатом округления чисел 17.5 и 18.5 является число 18. Авторы считают, что знание этих нюансов делает задачу слишком сложной...

Для округления вниз можно использовать:

- функции `trunc` в Паскале или `floor` в C++;
- изменение режима округления с помощью процедуры `SetRoundMode` (в Паскале);
- хранение вычисляемых данных в копейках. Ограничения задачи позволяют отвести для этого 64-битный целочисленный тип. Теперь комиссию, рассчитанную для каждой операции, следует перевести к соответствующему целочисленному типу.

Задача 4: Подматрицы

Конечно, же реализация одного запроса не представляет особой сложности для тех, кто работал с двумерными массивами. Она выполняется за время $O(M \times N)$. Если все запросы обрабатываются независимо друг от друга, то общая трудоёмкость этого алгоритма равна $O(M \times N \times Q)$.

Однако из-за больших ограничений на M , N , Q такое решение не укладывается в установленный лимит времени. Так, авторское решение, основанное на этом алгоритме, взяло только 35 тестов из 50.

Подходом для решения этой задачи за приемлемое время является т.н. *предпросчёт*. Его суть заключается в том, что мы рассчитываем вспомогательную информацию, а потом используем её для обработки запросов.

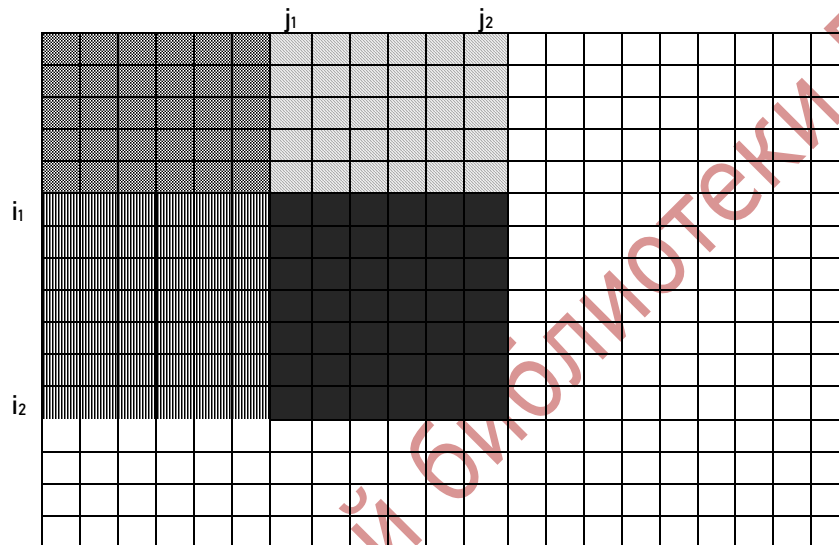
Рассчитаем вспомогательную матрицу $\mathbf{D}$ размером $M \times N$, где

$$d_{u,v} = \sum_{i=1}^u \sum_{j=1}^v a_{i,j}$$

Теперь для получения результата очередного запроса нам достаточно вычислить

$$d_{i_2,j_2} - d_{i_2,j_1-1} - d_{i_1-1,j_2} + d_{i_1-1,j_1-1}$$

Объяснение этого механизма показано на следующем рисунке (значения верхнего левого прямоугольника вычитаются дважды, и нам необходимо выполнить коррекцию):



Такой алгоритм имеет трудоёмкость $O(M \times N + Q)$, что достаточно для решения задачи.

Задача 1: Левая рука тьмы

Самая простая задача – 96 полных решений.

Рассмотрим два подхода к решению задачи. Первый годится для тех, кто знаком с контейнерами для хранения пар «ключ – значение» (такого рода контейнеры есть во всех разрешённых системах программирования, за исключением Free Pascal и Delphi).

Создаём контейнер `map <string, pair <int, int> >`. Ключом в этом контейнере будет имя индивида, а значение состоит из количества детей, для которых он был соответственно отцом и матерью.

В каждой записи о рождении выделяем имена отца, матери и ребёнка. После этого добавляем в контейнер новый элемент или (при его наличии) корректируем значение неключевой части.

После завершения чтения входных данных достаточно последовательно просмотреть содержимое контейнера и вывести хранящуюся в нём информацию.

Те участники, кто не знаком с ассоциативными контейнерами, могут воспользоваться вторым подходом.

Создаём массив из $3 * N$ элементов, содержащих имя индивида и признак его роли (отец, мать, ребёнок) в каждой записи о рождении. После чтения входных данных сортируем этот массив по именам. Тогда записи для одного индивида образуют группу из расположенных подряд элементов.

Последовательно просматриваем полученный массив, считая для каждой группы число записей, в которых индивид считается отцом и матерью соответственно. При изменении имени, а также после просмотра всего массива следует выводить информацию о соответствующем индивиде.

Авторское решение на Delphi, основанное на этом подходе, также показало удовлетворительный результат.

Трудоёмкость обоих подходов – $O(N * \log N)$.

Задача 2: Галлеоны, сикли и кнаты

На взгляд авторов – самая простая задача, хотя на самом деле она таковой не явилась ☹

Подводный камень заключается в том, что если пересчёт в галлеоны и сикли после выполнения каждой операции выполняется N раз, это может привести к превышению лимита времени. Поэтому лучше выполнить перевод каждой суммы в кнаты при вводе, а результаты вычислений хранить

в кнатах. Обратный перевод в галлеоны и сикли выполняется один раз, перед выводом результата.

Задача 3: Дребезжащие клавиши

Задача оказалась второй по сложности, как и предполагали авторы при её подготовке...

Введение подтестов для задачи обусловлено нежеланием принимать решения, в которых идёт вывод одного и того же ответа без анализа входных данных.

Пусть для анализа представлены две строки A и B. Если длина первой меньше длины второй – это очевидная опечатка. В противном случае просматриваем обе строки параллельно, анализируя символы с индексами i_A и i_B соответственно.

Если эти символы равны, увеличиваем оба индекса. В противном случае смотрим, совпадают ли текущий и предыдущий символы строки B. Если $i_B = 1$ или $B[i_B] \neq B[i_B - 1]$, имеет место опечатка, иначе увеличиваем индекс i_B .

Алгоритм завершает работу, когда просмотрена вся строка B. Если к этому времени строка A также просмотрена, мы определили ситуацию с дребезгом, в противном случае тоже обнаружили опечатку (последние символы первой строки отсутствуют во второй).

В ходе олимпиады было задано множество вопросов: как рассматривать ситуацию, когда строки A и B совпадают. В этом случае дребезга клавиш нет, как нет и опечатки. Это показано в первом подтесте примера.

Задача 4: Закупка урожая

Задача на аккуратную работу с двумерными массивами. Неожиданно она оказалась самой сложной... Видимо сказывается то, что массивы (а особенно двумерные массивы) исключены из школьной программы. Кроме того, для этой задачи нет авторского решения на Python, проходящего все тесты. Нет также ни одного полного решения на этом языке и у участников олимпиады. Что же, язык медленный, хотя и удобный для начального изучения...

Дальнейшее объяснение будет выполнено для языка C++.

Создаём контейнер `map <long long, long long>`, ключом в котором будет величина $a_{ij} + T * (|r - i| + |c - j|)$ для всех возможных i и j – цена закупки с учётом расходов на доставку, а значением – количество зерна, которое можно купить по этой цене.

После этого последовательно просматриваем этот контейнер и набираем количество купленного зерна до тех пор, пока склад не заполнится (или до конца контейнера).

Для участников, не знакомых с понятием ассоциативных контейнеров можно, как и в первой задаче, использовать массив размером $M * N$, элементами которого являются пары «стоимость тонны зерна с учётом доставки – количество продаваемого зерна» для каждого участка. После сортировки этого массива по стоимости просматриваем его последовательно (закупая всё предложенное зерно) до тех пор, пока количество закупленного зерна не достигнет Q .

Трудоёмкость обоих этих алгоритмов – $O(M * N * \log(M * N))$.

из электронной библиотеки БГУ

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ	3
УСЛОВИЯ ЗАДАЧ	5
2016/17 учебный год	5
Задача 1: Юные программисты	5
Задача 2: За верность игре	5
Задача 3: Дроби	7
Задача 4: Считаем слова	7
2017/18 учебный год	9
Задача 1: Наведение порядка	9
Задача 2: Первый опыт шифрования	11
Задача 3: Максимальный среди минимальных	12
Задача 4: Преобразование последовательности	12
2018/19 учебный год	14
Задача 1: Хорошие пароли	14
Задача 2: Экспрессные маршруты - 2	15
Задача 3: Заполнить матрицу	16
Задача 4: Железо любит ласку, чистоту и смазку...	17
2019/20 учебный год	19
Задача 1: За верность игре - 2	19
Задача 2: Казино	21
Задача 3: Расширение строки	21
Задача 4: Две скважины	22
2020/21 учебный год	24
Задача 1: Железнодорожная изгородь - 2	24
Задача 2: Непокрывающиеся интервалы	25
Задача 3: «Чужие» банкоматы	26
Задача 4: Подматрицы	27
2021/22 учебный год	29
Задача 1: Левая рука тьмы	29
Задача 2: Галлеоны, сикли и кнаты	30
Задача 3: Дребезжащие клавиши	31
Задача 4: Закупка урожая	32
РАЗБОР ЗАДАЧ	34
2016/17 учебный год	34
Задача 1: Юные программисты	34
Задача 2: За верность игре	34
Задача 3: Дроби	35
Задача 4: Считаем слова	35
2017/18 учебный год	36
Задача 1: Наведение порядка	36
Задача 2: Первый опыт шифрования	36

Задача 3: Максимальный среди минимальных	37
Задача 4: Преобразование последовательности	37
2018/19 учебный год	38
Задача 1: Хорошие пароли	38
Задача 2: Экспрессные маршруты - 2	38
Задача 3: Заполнить матрицу	39
Задача 4: Железо любит ласку, чистоту и смазку...	40
2019/20 учебный год	41
Задача 1: За верность игре – 2	41
Задача 2: Казино	41
Задача 3: Расширение строки	42
Задача 4: Две скважины	43
2020/21 учебный год	44
Задача 1: Железнодорожная изгородь – 2	44
Задача 2: Непокрывающиеся интервалы	44
Задача 3: «Чужие» банкоматы	45
Задача 4: Подматрицы	45
2021/22 учебный год	47
Задача 1: Левая рука тьмы	47
Задача 2: Галлеоны, сикли и кнаты	47
Задача 3: Дребезжащие клавиши	48
Задача 4: Закупка урожая	48

Учебное издание

Кашкевич Сергей Иванович

**СБОРНИК
ОЛИМПИАДНЫХ ЗАДАЧ
ПО ИНФОРМАТИКЕ**

**Практикум для студентов
факультета прикладной математики и информатики**

В пяти частях

Часть 3

В авторской редакции

Ответственный за выпуск *С. И. Кашкевич*

Подписано в печать 03.06.2022. Формат 60×84/16. Бумага офсетная.
Усл. печ. л. 3,02. Уч.-изд. л. 2,36. Тираж 50 экз. Заказ _____

Белорусский государственный университет.
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий № 1/270 от 03.04.2014.
пр. Независимости, 4, 220030, Минск.

Отпечатано с оригинал-макета заказчика
на копировально-множительной технике
факультета прикладной математики и информатики
Белорусского государственного университета.
пр. Независимости, 4, 220030, Минск.