

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ  
И ИНФОРМАТИКИ  
Кафедра информационных систем управления**

---

**С. И. Кашкевич, А. А. Толстиков**

# **СБОРНИК ОЛИМПИАДНЫХ ЗАДАЧ ПО ИНФОРМАТИКЕ**

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

**В пяти частях**

**Часть 1**

**МИНСК  
2016**

УДК 004(075.3)  
ББК 32.81к2  
К31

Утверждено на заседании кафедры  
информационных систем управления  
факультета прикладной математики и информатики  
17 декабря 2015 г., протокол № 6

Р е ц е н з е н т  
профессор кафедры компьютерных технологий и систем,  
доктор педагогических наук *В. В. Казачёнок*

**Кашкевич, С. И.**  
К31      Сборник олимпиадных задач по информатике : практикум.  
В 5 ч. Ч. 1/ С. И. Кашкевич, А. А. Толстиков. – Минск : БГУ, 2016.  
– 29 с.

Практикум содержит условия, а также разборы решений задач школьных районных олимпиад по информатике г. Минска (2013–2015 годы).

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., Толстиков А. А., 2016  
© БГУ, 2016

## 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 пособие. Кроме этого, авторы считают, что решение олимпиадных задач по информатике позволяет развивать логическое мышление и находить оптимальные решения даже в самых повседневных ситуациях.

Авторы предлагаемого вашему вниманию практикума в течение многих лет составляют 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. Их можно скачать из электронной библиотеки БГУ по следующим адресам:

[elib.bsu.by/handle/123456789/41655](http://elib.bsu.by/handle/123456789/41655) (часть I)

[elib.bsu.by/handle/123456789/48379](http://elib.bsu.by/handle/123456789/48379) (часть II)

[elib.bsu.by/handle/123456789/104667](http://elib.bsu.by/handle/123456789/104667) (часть III)

Однако с этого времени проводились всё новые соревнования, и авторы решили продолжить свою работу. По задумкам авторов, может быть выпущено пять частей новой книги, первая из которых предлагается вниманию читателей. В настоящее время (февраль 2016 года) идёт работа над второй частью, куда войдут материалы VII и VIII командных чемпионатов (2014 – 2015 годы).

В первой части сборника авторы собрали условия заданий второго этапа школьной республиканской олимпиады по учебному предмету «Информатика» в г. Минске (всего – 12 задач), предложенные в течение 2013 – 2015 годов, а также показали свои 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 не рассмотренные случаи граничных значений параметров.

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.

Авторы благодарят профессора кафедры компьютерных технологий и систем, доктора педагогических наук В.В. Казачёнка за рецензирование рукописи. Благодарим также студентов факультета прикладной математики и информатики Э.О. Гамисония, С.А. Коноплича и А.Э. Турчиновича, внимательно прочитавших рукопись.

Тесты к задачам можно скачать из интернета по ссылке <https://yadi.sk/d/anmhzNRtp65KF>

Замечания и предложения по усовершенствованию этой работы просим присылать на электронные адреса [kash@bsu.by](mailto:kash@bsu.by) или [tolstikov@bsu.by](mailto:tolstikov@bsu.by).

## УСЛОВИЯ ЗАДАЧ

2013/14 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: Испытания поезда

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Новый поезд проходит заключительный этап приёмочных испытаний. Одно из заданий этого этапа предполагает проезд от станции А до станции В, расстояние между которыми равно  $L$ , в следующем режиме:

разогнаться с ускорением  $a_1$  до скорости, не превосходящей  $v$ ;

проехать с этой скоростью время, не меньшее  $t$ ;

остановиться на станции В, тормозя с замедлением  $a_2$ .

Рассчитайте минимальное время, которое поезд будет находиться в пути во время выполнения этого задания.

Формат входных данных. Единственная строка содержит числа  $L$ ,  $a_1$ ,  $a_2$ ,  $v$ ,  $t$ . Величины  $L$ ,  $a_1$ ,  $a_2$ ,  $v$  — положительные, действительные;  $t$  — неотрицательная, действительная. Дробная часть содержит не более 4 десятичных цифр, а целая часть не превосходит  $10^3$ . В 50 % тестов величина  $t$  равна нулю.

Формат выходных данных. Выведите единственное число — искомое время нахождения поезда в пути, рассчитанное с точностью до  $10^{-4}$ .

Примеры входных и выходных данных

40.5 0.5 0.6 3 7.5	19.00000000
20.5 0.5 0.6 3 7.5	14.37300711

### Задача 2: Место для фабрики

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Компания, в которой Вы проходите стажировку в качестве аналитика, планирует начать разработку крупного месторождения ценной руды. План месторождения представляет собой прямоугольник размером  $M \times N$ , разбитый на единичные квадраты — участки. Геологи оценили

запасы руды на каждом из участков. Прежде чем приступить к разработке месторождения, на территории месторождения необходимо построить обогатительную фабрику. Под площадку для постройки фабрики необходимо отвести квадрат из девяти смежных участков, и добыча руды на этом месте будет невозможна. Вам поручено первое задание — определить такое место для фабрики, при котором количество не добытой из-за её постройки руды будет минимальным.

Формат входных данных. Первая строка содержит два числа  $M$  и  $N$  ( $3 \leq M, N$ ;  $M \cdot N \leq 10^6$ , в 80 % тестов величины  $M$  и  $N$  не превосходят 100). Далее следуют  $M$  строк из  $N$  чисел каждая — разведанные запасы руды на каждом участке. Эти числа целые, неотрицательные, не превосходящие 100000.

Формат выходных данных. Выведите единственное число — запасы руды на площадке, которую Вы определили для постройки фабрики.

#### Примеры входных и выходных данных

5 7 10 2 3 7 10 4 8 3 2 1 9 6 2 1 0 3 6 7 8 9 10 5 4 3 0 2 1 8 9 2 3 10 6 4 8	27
--	----

### **Задача 3: Набрать SMS**

Ограничение по времени: 0.25 секунды

Ограничение по памяти: 16 мегабайт

SMS-сообщение представляет собой строку, содержащую прописные и строчные латинские буквы, цифры, пробел, точку, запятую и знак «плюс». Для набора сообщения на старых кнопочных мобильных телефонах используются клавиши с цифрами и клавиша «#». За цифровыми клавишами закреплены символы, которые можно ввести в сообщении:

клавиша	символы		клавиша	символы
1	.,1		7	PQRS7
2	ABC2		8	TUV8
3	DEF3		9	WXYZ9

<i>клавиша</i>	<i>символы</i>		<i>клавиша</i>	<i>символы</i>
4	GHI4		0	+0
5	JKL5		#	пробел
6	MNO6			

Кроме того, однократное нажатие клавиши «\*» позволяет переключаться между вводом прописных и строчных букв. Для того, чтобы набрать символ из сообщения, необходимо нажать соответствующую клавишу столько раз, каким по порядку находится нужный символ в списке символов, закреплённых за клавишей. Так, для набора буквы R потребуется нажать клавишу «7» три раза, для набора цифры 9 соответствующую клавишу необходимо нажать пять раз. Для того, чтобы последовательно набирать символы, закреплённые за одной клавишей и записанные в одном регистре (прописные либо строчные), необходимо после ввода предыдущего символа выдержать паузу, и только потом начинать ввод следующего символа. Так для ввода строки NO необходимо дважды нажать клавишу «6», выдержать паузу, а затем трижды нажать эту же клавишу. Предположим, что на однократное нажатие клавиши тратится одна единица времени, необходимая пауза длится три единицы времени, а на перенос пальца с одной клавиши на другую надо потратить две единицы времени. Сможете ли Вы рассчитать минимальное количество времени, которое займёт ввод с клавиатуры всего сообщения? Отсчёт времени начинается с первого нажатия любой клавиши и заканчивается в момент набора всего сообщения.

Формат входных данных. Первая строка файла содержит число 1 или 2 — признак того, какой режим ввода букв включён изначально (1 — прописные, 2 — строчные). Во второй строке записан текст сообщения — непустая строка длиной не более 30000 символов (в 50 % тестов длина строки не превосходит 255 символов). Кроме того, в 20 % тестов текст сообщения содержит только латинские буквы, записанные в одном регистре.

Формат выходных данных. Выведите единственное число — время, затраченное на набор сообщения.

### Примеры входных и выходных данных

1 HELLO	22
2 My phone is +375286198121	127
2 MaMa	22

### **Задача 4: Дроби**

Ограничение по времени: 3 секунды

Ограничение по памяти: 64 мегабайта

Найдите количество правильных несократимых дробей, не превосходящих  $X$ , знаменатель которых не превосходит  $N$ . Напомним, что дробь называется *правильной*, если её числитель и знаменатель – натуральные числа, и числитель меньше знаменателя.

Формат входных данных. Первая строка содержит значение  $X$  ( $0 < X \leq 1$ ), записанное не более чем с 4 десятичными цифрами в дробной части. Во второй строке записана величина  $N$  ( $2 \leq N \leq 100000$ ). В 20 % тестов эта величина не превосходит 10, в 60 % тестов – 1000, а в 90 % тестов – 10000.

Формат выходных данных. Выведите единственное число — количество найденных дробей.

### Примеры входных и выходных данных

0.4 6	5
0.6543 10	20



## 2014/15 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: The king and the knight

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 мегабайт

Напомним прежде всего некоторые шахматные правила, которые понадобятся для решения задачи.

Король (king) за один ход может перейти из той клетки шахматной доски, в которой он находится, в любую соседнюю по вертикали, горизонтали или диагонали. Естественно, он не может при этом выйти за пределы доски.

Конь (knight) может пойти на любое поле доски, если оно располагается на другом конце русской буквы Г (то есть вначале конь перемещается на два поля по горизонтали или по вертикали, а затем на одну клетку перпендикулярно первоначальному направлению). Выходить за границы доски также нельзя.

Будем говорить, что фигура *A* находится под боем фигуры *B*, если фигура *B* за один ход может переместиться в клетку, в которой находится фигура *A*.

А теперь – задача...

В различных клетках шахматной доски находятся две фигуры: король и конь. Вам требуется выяснить истинность или ложность каждого из четырёх утверждений:

- 1) король находится под боем коня;
- 2) король не находится под боем коня, но будет находиться под боем после одного допустимого хода коня;
- 3) конь находится под боем короля;
- 4) конь не находится под боем короля, но будет находиться под боем после одного допустимого хода короля

Сможете решить эту задачу? За правильный ответ на каждый вопрос начисляется 0.5 балла, так что за полный ответ на тест можно получить 2 балла.

Формат входных данных. В первой строке файла записывается положение короля в стандартной шахматной нотации (согласно этой нотации, строки обозначаются снизу вверх цифрами от 1 до 8, а столбцы сле-

ва направо латинскими буквами от **a** до **h**, так что нижняя левая клетка обозначается как **a1**, а верхняя правая – **h8**). Во второй строке аналогичным образом записывается положение коня.

Формат выходных данных. Выведите строку из четырёх символов, каждый из которых принимает значение **Y** или **N** в зависимости от того, истинно или ложно очередное утверждение.

#### Примеры входных и выходных данных

a1 b2	NNYN
a1 b3	NNNY

## **Задача 2: Слова-перевёртыши**

Ограничение по времени: 3 секунды

Ограничение по памяти: 64 мегабайта

Вам, наверное, известно, что некоторые символы, записанные на листе бумаги, после поворота этого листа на 180° также можно прочитать, хотя и по-другому. Так, прописные латинские буквы **H, I, N, O, S, X, Z** и цифра **0** после поворота читаются так же, а буква **M** и цифра **6** «превращаются» соответственно в символы **W** и **9**, и наоборот.

Непустую строку, состоящую из прописных латинских букв и цифр, будем называть *словом-перевёртышем*, если она составлена только из перечисленных выше символов. После поворота листа это слово также можно прочитать, хотя и по-другому. Так, строка **HOW** «превращается» в строку **MOH**. С другой стороны, строка **HOME** словом-перевёртышем не является.

Найдите все слова-перевёртыши во входном файле.

Формат входных данных. Первая строка входного файла содержит величину  $K$  ( $1 \leq K \leq 1000$ ) – количество обрабатываемых в дальнейшем строк.

Каждая из последующих  $K$  строк состоит только из прописных латинских букв и цифр, пустых строк нет. Длина одной строки не превосходит 3000 символов (в 50 % тестов эта величина не превосходит 255).

Формат выходных данных. В первой строке выведите одно число – количество найденных слов-перевёртышей. Каждая из последующих строк должна содержать строку, которая получится из очередного слова-

перевёртыша после поворота листа бумаги. Слова-перевёртыши нужно выводить в том порядке, в котором они встречаются во входном файле.

В начале и конце каждой строки не должно быть пробелов!

#### Примеры входных и выходных данных

4 LOL WOW WIHO9 PI	2 MOM 6OHIM
--------------------------------	-------------------

### **Задача 3: Разрисованная плоскость**

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 мегабайт

Некоторые из точек координатной плоскости с целочисленными координатами раскрашены в один из  $N$  цветов по следующему принципу: точки, лежащие на линии, определяемой уравнением

$$y = x + k,$$

где  $k \geq 0$ , выкрашены в цвет с номером  $1 + k \bmod N$ . Точки, лежащие на линиях, для которых  $k < 0$ , считаются не закрашенными.

На этой плоскости нарисован прямоугольник со сторонами, параллельными осям координат. Левая верхняя вершина этого прямоугольника имеет целочисленные координаты  $(x_1, y_1)$ , а правая нижняя –  $(x_2, y_2)$ .

Определите, сколько точек каждого цвета находятся внутри и на границе нарисованного прямоугольника.

Формат входных данных. Первая строка входного файла содержит значение  $N$  ( $1 \leq N \leq 1000$ ), а во второй строке записаны величины  $x_1, y_1, x_2, y_2$  (числа, не превосходящие по модулю  $10^7$ ,  $x_1 < y_1, x_2 < y_2$ ). В 20 % тестов суммарное число точек внутри и на границе прямоугольника не превосходит 100, а в 80 % тестов это число не превосходит  $10^6$ .

Формат выходных данных. Выведите в одну строку  $N$  чисел – количество точек, находящихся внутри и на границе прямоугольника и раскрашенных в соответствующий цвет. Числа должны быть разделены единственным пробелом. Пробелы в начале и конце строки не допускаются.

### Примеры входных и выходных данных

5 -2 4 1 -1	5 5 4 4 3
10 0 0 1 -1	1 0 0 0 0 0 0 0 0 0

### **Задача 4: Подняться на лестницу**

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

При прохождении компьютерной игры Вася дошел до финального испытания. Ему необходимо взобраться на вершину пирамиды. Вокруг пирамиды идет лестница, однако за долгие годы некоторые из ступенек этой лестницы стали слишком опасными для того, чтобы можно было на них наступать.

Вася очень аккуратно проходил все предыдущие испытания, и поэтому у него есть все подсказки о том, на какие ступеньки наступать нельзя.

Лестница состоит из  $N$  ступенек, пронумерованных от 1 до  $N$  от основания пирамиды до её вершины, и в ней  $K$  опасных ступенек. Кроме того, персонаж Васи за одну секунду может сделать один шаг и подняться на 1, 2, ...,  $S$  ступенек.

До прохождения испытания персонаж стоит перед первой ступенькой, а для успешного его прохождения требуется оказаться на  $N$ -ой ступеньке.

Определите минимальное и максимальное время, за которое Вася может пройти финальное испытание.

Формат входных данных. В первой строке входного файла записаны три целых числа  $N$ ,  $K$  и  $S$  ( $1 \leq N \leq 200000$ ,  $1 \leq S < N$ ,  $0 \leq K < N$ ). Если  $K > 0$ , то во второй строке записаны  $K$  различных целых чисел  $a_i$  ( $1 \leq a_i < N$ ) – номера опасных ступенек. Гарантируется, что можно подняться на вершину пирамиды, не наступая на опасные ступеньки. В 80 % тестов  $N \leq 1000$ .

Формат выходных данных. В единственной строке выходного файла выведите два целых числа: минимальное и максимальное время, за которое Вася может пройти финальное испытание.

Примеры входных и выходных данных

10 0 3	4 10
6 2 2 5 2	4 4

из электронной библиотеки БГУ

## 2015/16 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: Районная олимпиада по информатике

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

В районной олимпиаде по информатике (втором этапе республиканской олимпиады) участвуют команды из  $N$  районов. Команда  $i$ -го района состоит из  $P_i$  участников, а квота для участия в третьем этапе, выделенная этому району до начала соревнований, составляет  $Q_i$  человек. Однако эта квота может быть увеличена за счёт участников, набравших такое же количество баллов, что и участник с рангом  $Q_i$ . Так, если квота для какого-нибудь района составляет три участника, а пять лучших участников этого района набрали соответственно 286, 285.5, 285.5, 285.5, и 274 балла, то на третий этап олимпиады проходят четверо лучших участников второго этапа из этого района.

Кроме того, участник, не набравший ни одного балла на втором этапе олимпиады, не проходит на третий этап, даже если в квоте района остались свободные места.

Каждому участнику второго этапа предложено для решения четыре задачи. Максимальная оценка по каждой задаче составляет 100 баллов, причём баллы по задачам могут считаться с точностью до сотых долей.

Вам необходимо на основании информации об официальных результатах второго этапа олимпиады составить таблицу участников третьего этапа.

Формат входных данных. Первая строка содержит величину  $N$  ( $2 \leq N \leq 200$ ). Во второй строке записаны  $N$  чисел — величины  $P_i$  ( $2 \leq P_i \leq 1000$ ,  $P = \sum_{i=1}^N P_i \leq 100000$ , для 50 % тестов  $P \leq 10000$ ). Третья строка содержит величины  $Q_i$  в таком же формате ( $0 < Q_i < P_i$ ).

Далее следуют  $P$  строк с результатами каждого участника олимпиады. Каждая из этих строк содержит шесть чисел: уникальный код участника (целое положительное число, не превосходящее  $10^6$ ), номер команды (целое число в интервале от 1 до  $N$ ), баллы, набранные за решения каждой задачи (неотрицательные числа).

Никакой сортировки строк с результатами соревнований нет. Тесты подобраны таким образом, что на третий этап олимпиады должен

проходить хотя бы один школьник. Кроме того, в 40 % тестов баллы по задачам считаются с точностью до целых.

Формат выходных данных. В первой строке выведите общее количество участников третьего этапа  $Q$ .

Каждая из последующих  $Q$  строк содержит информацию об участниках третьего этапа и включает два числа: код участника и сумму баллов, набранных на втором этапе соревнований. Строки файла должны быть отсортированы по убыванию суммы баллов, а при равенстве этого показателя — по возрастанию кода участника.

Примеры входных и выходных данных

5 4 2 3 3 3 2 1 1 2 1 101 1 12 86 100 100 102 1 13.48 15.97 100 38 403 4 11 86 98 0 201 2 0 0 5 0 104 1 57 35 91 95 202 2 48 11.95 99.99 30 103 1 37 28 93 47 401 4 91 41 100 100 301 3 0 5 0 0 303 3 2 0 100 10 302 3 47 71 100 78 402 4 18 32 100 74 503 5 55.16 95 100 65.18 502 5 100 100 100 15.34 501 5 0 18 0 0	8 401 332.00 502 315.34 503 315.34 101 298.00 302 296.00 104 278.00 402 224.00 202 189.94
5 4 2 3 3 3 2 1 1 2 1 101 1 12 86 100 100 102 1 13.48 15.97 100 38 403 4 0 0 0 0 201 2 0 0 5 0 104 1 57 35 91 95 202 2 48 11.95 99.99 30 103 1 37 28 93 47 401 4 0 0 0 0 301 3 0 5 0 0	6 503 315.34 101 298.00 302 296.00 104 278.00 402 224.00 202 189.94

303	3	2	0	100	10	
302	3	47	71	100	78	
402	4	18	32	100	74	
503	5	55.16	95	100	65.18	
502	5	45	98	100	11.07	
501	5	0	0	0	0	

Замечание. В первом примере квота района с номером 5 увеличивается на одного участника, а во втором — район с номером 4 лишается одного участника.

## Задача 2: Чекер для sudoku

Ограничение по времени: 1 секунда

Ограничение по памяти: 64 мегабайта

Судуку — головоломка, получившая невероятную популярность, начиная с 1980-х — 1990-х годов. Игровое поле для этой головоломки представляет собой квадрат размером  $9 \times 9$ , разделённый на меньшие квадраты со стороной в 3 клетки. Таким образом, всё игровое поле состоит из 81 клетки. В них уже в начале игры стоят некоторые числа (от 1 до 9), называемые *подсказками*. От игрока требуется заполнить свободные клетки цифрами от 1 до 9 так, чтобы в каждой строке, в каждом столбце и в каждом малом квадрате  $3 \times 3$  каждая цифра встречалась бы только один раз.

4	1	3				2	5	
		2	3					
			4	9			8	
7	8	9			3			
5			9		6			1
			2			9	7	5
	4			3	7			
					5	4		
	5	1				3	6	7

4	1	3	7	6	8	2	5	9
8	9	2	3	5	1	7	4	6
6	7	5	4	9	2	1	8	3
7	8	9	5	1	3	6	2	4
5	2	4	9	7	6	8	3	1
1	3	6	2	8	4	9	7	5
9	4	8	6	3	7	5	1	2
3	6	7	1	2	5	4	9	8
2	5	1	8	4	9	3	6	7

На приведенном рисунке показан пример исходного заполнения игрового поля подсказками и решённая головоломка.



Правильно составленная головоломка имеет только одно решение. Тем не менее, иногда предлагаются варианты sudoku с несколькими вариантами решения.

Соревнования по решению sudoku предполагают, что каждый участник получает в качестве задания информацию о подсказках и сдаёт в качестве результата полностью заполненное цифрами игровое поле. Проверка правильности решения выполняется автоматически.

К сожалению, в случае наличия нескольких вариантов решения проверка путём сравнения с эталонным решением является недостаточной. Для проверки правильности необходимо написать специальную программу — чекер. Вам требуется написать такую программу...

Чекер должен находить следующие ошибки в решении предложенной головоломки:

- наличие в решении символов, не совпадающих с цифрами от 1 до 9;
- наличие одинаковых цифр в столбцах, строках, малых квадратах;
- наличие цифр, стоящих на месте подсказок и не совпадающих с подсказками.

Формат входных данных. В первой строке задаётся величина  $Q$  ( $2 \leq Q \leq 10$ ) — количество решений одной головоломки, каждое из которых должен проверить чекер. Следующие девять строк входного файла содержит 9 символов каждая и содержат описание исходной головоломки. В этом описании допустимы символы '1', ..., '9', соответствующие цифрам-подсказкам, и символ '\*', означающий незаполненную клетку.

После этого следуют описания каждого решения (9 строк по 9 символов для одного решения). В описании решений могут встретиться только символы с кодами от 32 до 127.

Формат выходных данных. Выведите строку из  $Q$  символов 'Y' или 'N'. Символ 'Y' означает, что очередное решение — правильное, в противном случае чекер обнаружил хотя бы одну ошибку.

#### Пример входных и выходных данных

3 413***25* **23***** ***49**8* 789**3*** 5**9*6**1 ***2**975 *4**37***	YNN
--	-----

*****54**	
*51***367	
413768259	
892351746	
675492183	
789513624	
524976831	
136284975	
948637512	
367125498	
251849367	
213768254	
892351746	
675492183	
789513624	
524976831	
136284975	
948637512	
467125398	
251849367	
4137a8259	
89235b746	
6754921c3	
789*13624	
52*976831	
136254*75	
48637512	
3 7125498	
25 849367	

### Задача 3: Железнодорожная изгородь

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 мегабайт

Метод «железнодорожной изгороди»<sup>1</sup> — 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...

<sup>1</sup> По правде говоря, авторы задачи не понимают смысл этого термина ☺

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

Вам требуется зашифровать сообщение методом «железнодорожной изгороди».

Формат выходных данных. Выведите единственную строку с зашифрованным сообщением.

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

Антон и Борис увлечены новой игрой, суть которой заключается в следующем. Один из игроков, по жребию, становится ведущим. Ведущий придумывает четырёхзначное слово (возможно, с нулями в старших

разрядах) и передаёт ход другому игроку. В дальнейшем игроки делают ходы по очереди.

Выполняя ход, игрок может увеличить одну из цифр текущего числа на 1, 2 или 3. Естественно, новое значение этой цифры не должно превосходить девяти. Игрок, получивший в результате хода число 9999, проигрывает.

Определите, выиграет ли ведущий игрок при заданном начальном числе, если в процессе игры ни Антон, ни Борис не будут совершать ошибок.

Формат входных данных. В первой строке записано количество подтестов  $Q$  ( $3 \leq Q \leq 10$ ). В каждой из последующих строк записано одно число в диапазоне от 1 до 9998 — очередное придуманное ведущим игроком число. Незначащие нули в этих числах не записываются.

Для 10 % тестов во всех их подтестах десятичная запись придуманных чисел содержит три девятки.

Формат выходных данных. Выведите строку из  $Q$  символов. В очередной позиции этой строки записывается символ 'Y', если ведущий игрок выигрывает, и 'N' — в противном случае.

Примеры входных и выходных данных

5 9989 9995 9696 8999 8899	YNYYN
4 9989 999 9899 5999	YYYN

## РАЗБОР ЗАДАЧ

2013/14 учебный год

### Задача 1: Испытания поезда

Решение задачи состоит в разборе двух случаев:

- поезд может разогнаться до максимально допустимой скорости  $v$  и ехать с этой скоростью время, не меньшее, чем  $t$ ;
- из-за малого расстояния между станциями скорость поезда будет меньше, чем  $v$ , а ехать с этой скоростью он будет ровно  $t$ .

В первом случае поезд разгонится до скорости  $v$  за время, равное  $t_1 = \frac{v}{a_1}$ , и пройдёт за это время расстояние  $S_1 = \frac{v^2}{2a_1}$ . Аналогичным образом рассчитывается время, затрачиваемое на торможение, и расстояние, пройденное за это время:  $t_2 = \frac{v}{a_2}$ ,  $S_2 = \frac{v^2}{2a_2}$ . Если величина  $t_3 = \frac{(L-S_1-S_2)}{v}$  больше либо равна  $t$ , искомое время рассчитывается по формуле

$$t_1 + t_2 + t_3.$$

Во втором случае необходимо рассчитать максимальную скорость  $u$ , до которой может разогнаться поезд. Эта скорость рассчитывается исходя из квадратного уравнения

$$\frac{u^2}{2a_1} + \frac{u^2}{2a_2} + ut = L$$

По теореме Виета один из этих корней – положительный, а другой – отрицательный. После этого сводим задачу к первому случаю, подставляя вместо  $v$  найденное положительное значение  $u$ .

### Задача 2: Место для фабрики

Для решения задачи требуется корректно организовать работу с двумерными массивами.

Пусть `Field` – двумерный массив с исходными данными. Тогда стоимость не извлечённой руды, если центр фабрики стоит в позиции  $[i, j]$ , вычисляется по формуле

```
Sum := Field[i-1, j-1] + Field[i-1, j] +  
Field[i-1, j+1] + Field[i, j-1] +  
Field[i, j] + Field[i, j+1] + Field[i+1, j-1] +  
Field[i+1, j] + Field[i+1, j+1];
```

После этого остаётся рассчитать минимум этого значения для всех допустимых  $i$  и  $j$ .

### Задача 3: Набрать SMS

По мнению авторов, это – самая простая задача в наборе. Для её решения требуется лишь правильно определять события «номер символа на клавише», «пауза» и «перенос пальца на другую клавишу» и увеличивать время набора SMS на соответствующее слагаемое.

### Задача 4: Дроби

Первый подход к решению этой задачи основан на построении начальной части ряда Фарея<sup>2</sup> порядка  $N$ . Напомним, что ряд Фарея порядка  $N$  представляет собой возрастающий ряд всех положительных несократимых правильных дробей, знаменатель которых меньше или равен  $N$ . Приведём пример этого ряда для  $N = 6$ :

$$F_6 = \left\{ \frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{4}{5}, \frac{5}{6} \right\}$$

Алгоритм генерации ряда Фарея для заданного  $N$  можно записать следующим образом. Два первых элемента, очевидно, равны  $1/N$  и  $1/(N-1)$ . Каждый очередной элемент строится на основе двух предыдущих. Пусть последние построенные элементы равны соответственно  $a/b$  и  $c/d$ . Тогда числитель  $p$  и знаменатель  $q$  следующего элемента вычисляются по формулам

$$p = ((N+b) \operatorname{div} d) \cdot c - a, q = ((N+b) \operatorname{div} d) \cdot d - b$$

Вычисления прекращаются, когда очередной член ряда станет больше, чем  $X$ .

Опишем альтернативный подход к решению этой задачи. Пусть  $f(q)$  равно количеству несократимых дробей со знаменателем  $q$ , не превосходящих  $X$ . Тогда, легко видеть, что  $f(1) = 1$ , если  $X = 1$  и  $f(1) = 0$  иначе. Остальные значения  $f(q)$  можно вычислить, используя рекуррентное соотношение:

$$f(q) = [X * q] - \sum_{1 < d | q} f(q/d).$$

При вычислении  $f(q)$  мы берем все дроби со знаменателем  $q$  и считаем из них те, у которых  $d$  равен наибольшему общему делителю числителя и знаменателя. Эти вычисления легко организовать, перебирая возможные делители, за время  $O(N * \log(N))$ .

---

<sup>2</sup> Джон Фарей (John Farey) — известный геолог, один из пионеров геофизики. Его единственным вкладом в математику были дроби, названные его именем. Ряд Фарея был построен в 1816 году.

### Задача 1: The king and the knight

Вначале опишем полное решение задачи. Условия должны проверяться не в том порядке, в котором они описаны в условии задачи, а в порядке, описанном ниже.

а) конь находится под боем короля, если модуль расстояния между этими фигурами по вертикали и горизонтали не превосходит единицы;

б) конь не находится под боем короля, но будет находиться под боем после одного допустимого хода короля, если не выполняется предыдущее условие, но модуль расстояния между этими фигурами по вертикали и горизонтали не превосходит двух;

с) король находится под боем коня, если модуль расстояния между этими фигурами по одному из направлений (вертикали либо диагонали) равен единице, а по противоположному направлению – двум;

д) для проверки последнего условия определяем все позиции, куда может попасть конь (количество таких позиций – от 2 до 8). Для каждой из таких позиций проверяем условие с).

Авторы намеренно упростили задачу, добавив в неё элемент угадывания. Так, при истинности самого простого условия а) можно сразу сказать, что условия б) и с) ложны, а вместо проверки условия д) вывести случайный ответ. В этом случае тест набирает от 1.5 до 2 баллов (как повезёт ☺). Аналогичное угадывание можно записать для истинности условия б).

Увы, такой возможностью воспользовалось небольшое количество участников...

### Задача 2: Слова-перевёртыши

Задача, требующая умения работать с двойными циклами и понимания, как можно выйти из внутреннего цикла по инструкции break.

Формируем две строки:

A = "HINOSXZ0M6"

B = "HINOSXZ0W9"

Для каждого символа с очередной строки входного файла определяем позицию  $i$  его вхождения в строку **A**. Если оказывается, что с не входит в строку **A**, прекращаем обработку очередной строки входного файла и переходим к следующей строке. В противном случае помещаем в очередную позицию выходной строки  $i$ -й символ строки **B**.

Ограничение на длину строки, указанное для 50 % тестов, позволяет дать неполную оценку задачи для участников, работающих в «чистой» системе программирования Free Pascal, где длина строки типа `string` не превосходит 255.

Конечно же, для дальнейшей работы в системе программирования Free Pascal (как в этой, так и в других задачах) следует указывать в начале каждой программы «волшебный» режим компиляции

```
{ $mode delphi }
```

который, в частности, позволяет увеличивать как размер стандартной строки, так и размер данных типа `integer`.

### Задача 3: Разрисованная плоскость

Первое из решений этой задачи, которое приходит в голову, заключается в следующем: для каждой точки  $(x, y)$ , лежащей внутри и на границе прямоугольника, вычислить значение

$$k = y - x$$

и увеличить счётчик точек соответствующего цвета. Это решение работает только на сниженных ограничениях (так, авторское решение, основанное на таком подходе, взяло 80 баллов из 100).

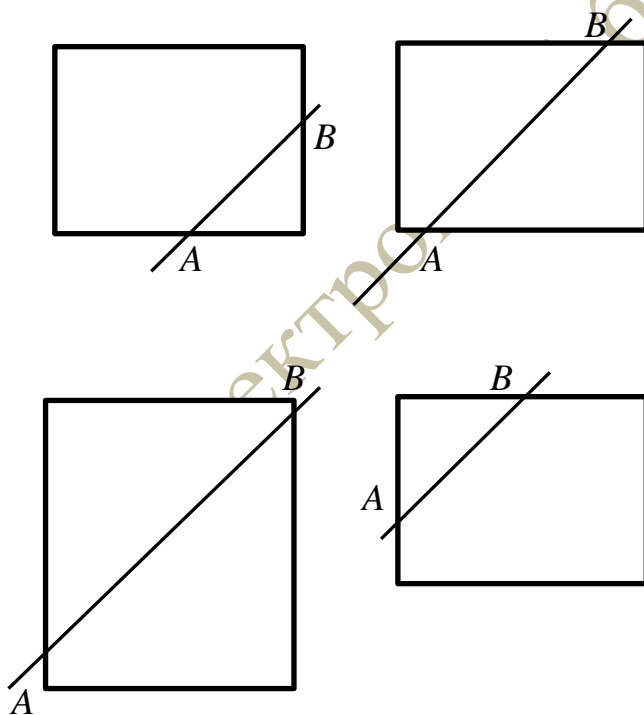


Рис. 1 Варианты пересечения прямой и прямоугольника

Полное решение задачи заключается в следующем. Прежде всего, определяем, не лежит ли прямоугольник ниже линии

$$y = x$$

(в этом случае разрисованных точек внутри и на границе нет).

После этого рассчитываем количество цветных линий, пересекающих прямоугольник. Эта величина равна

$$(y_1 - y_2 + 1) + (x_2 - x_1)$$

Для каждой из таких линий определяем её цвет и вариант пересечения с прямоугольником в соответствии с рисунком 1, после чего рассчитываем абсциссы точек  $A$  и  $B$ . Количество точек внутри и на границе прямоугольника, соответствующее



ющих обрабатываемой линии, равно  $x_B - x_A + 1$ .

Такой алгоритм, хоть и не является оптимальным, позволяет уложиться в заданное время для всех тестов. Авторы не захотели усложнять задачу, задавая бóльшие ограничения на размер прямоугольника.

#### **Задача 4: Подняться на лестницу**

Решение задачи основано на идеях динамического программирования. Похожая по идее (но не по условию) задача «Экспрессные маршруты» была предложена на районной олимпиаде по информатике в 2007/08 учебном году.

Максимальное время, за которое можно подняться по лестнице, очевидно, равно  $N - K$  и соответствует ситуации, когда Вася наступает на все ступеньки, за исключением опасных.

Для расчёта минимального времени введём функцию  $F(i)$  – минимальное время, за которое Вася может подняться на ступеньку с номером  $i$ . Оптимальным вариантом прохода будет такой: шагнуть на ступеньку с номером  $i$  с самой дальней из неопасных ступенек, отстоящих от неё не более чем на  $S$  (или с земли). Пусть  $p$  – номер такой ступеньки. Тогда формула для расчёта  $F(i)$  примет вид

$$F(0) = 0$$

$$F(i) = -1 \text{ для опасных ступенек}$$

$$F(i) = F(p) + 1 \text{ для безопасных ступенек}$$

Если каждый раз мы будем рассчитывать величину  $p$ , начиная от ступеньки  $i$ , то этот алгоритм будет иметь трудоёмкость  $O(N * S)$ . Однако можно заметить, что если на ступеньку  $i$  можно шагнуть со ступеньки  $p$ , то на ступеньку  $(i+1)$  можно будет ступить либо со ступеньки  $p$  либо со следующей неопасной ступеньки. Эта оптимизация позволяет реализовать вычисления описанного рекуррентного соотношения за линейное время.

Отметим, что эти же идеи позволяют показать, что «жадный» алгоритм, по которому мы каждый раз переходим на самую дальнюю неопасную ступеньку, также будет давать правильный ответ.

### Задача 1: Районная олимпиада по информатике

Задача требует от участника следующих умений:

- работы с составными типами данных (структурами);
- выполнения сортировки по нестандартному критерию;
- сравнения чисел с дробной частью.

Решение задачи сводится к следующим шагам. Вначале строится массив структур с полями «код участника» и «сумма набранных баллов». Этот массив сортируется согласно критерию, записанному в формате выходных данных. После этого отсортированный массив последовательно просматривается, и для каждого участника определяется его район. Если квота района не заполнена или может быть превышена за счёт этого участника, то он получает статус «проходит на следующий этап». Наконец, если окажется, что очередной участник не набрал ни одного балла, просмотр массива (и заполнение квот) прекращается.

Возможные подводные камни:

1) Поскольку числа с дробной частью хранятся в памяти компьютера с погрешностью, сумма четырёх чисел не совпадает с суммой, рассчитанной вручную. Так сумма чисел 48, 11.96, 99.99 и 30 не будет точно равна 189.95. Поэтому вместо точного сравнения необходимо считать, что суммы баллов считаются равными, если модуль их разности не превосходит 0.005. Другим подходом для решения этой проблемы является умножение количества набранных баллов на 100 и округления, такой переход к целым числам избавляет от учёта погрешности.

2) Сортировка с трудоёмкостью  $O(N^2)$  является слишком медленной и позволяет взять порядка 50 % баллов. Для полного решения задачи требуется применять сортировку с трудоёмкостью  $O(N \log N)$ .

### Задача 2: Чекер для sudoku

Задача на «реализацию», не требующая знания специфических алгоритмов, но предполагающая умение обрабатывать двумерные массивы.

После ввода информации об очередном решении вначале следует проверить, нет ли в нём недопустимых символов и нет ли символов, отличающихся от подсказок. Если эта проверка завершена успешно, переходим к следующему этапу. Строим булевский массив  $P$  из 9 элементов, значения которого определяют, была ли использована соответствующая цифра. Перед каждой проверкой этот массив заполняется значениями false.

Проверяем последовательно строки, столбцы, малые прямоугольники. Если элемент массива  $P$  с индексом, соответствующим очередному символу, равен `true`, это означает, что чекер нашёл ошибку в решении. В противном случае изменяем значение этого элемента на `true`.

Подводным камнем может быть наличие пробелов во входных данных. Участники олимпиады, работающие на языке C++, могут плохо прочитать строку, если они используют операцию `>>` вместо вызова функции `getline`.

### Задача 3: Железнодорожная изгородь

Самый простой, по мнению авторов, подход к решению этой задачи состоит в том, что на основе анализа входной строки  $S$  сразу же формируется выходная строка.

Первая строка изгороди соответствует позициям зашифрованного текста  $1, 2*(k-1)+1, \dots, 2*t*(k-1)+1$ , где  $t = \text{length}(S) \text{ div } k$ . Читателям предлагается вывести соответствующие формулы для последующих строк изгороди.

Такое решение имеет сложность  $O(\text{length}(S))$  по памяти и  $O(\text{length}(S))$  по времени.

Участники, использующие язык C++, могут использовать конструкцию вида

`vector < vector < int >> v(k);`

(аналог двумерной матрицы) и заполнять строки этой матрицы без пропусков. Этот подход имеет такую же трудоёмкость, но более прост в реализации, по мнению авторов.

Наконец, формирование изгороди в виде двумерного массива, как показано на рисунке в условии, является нерациональным, поскольку приводит к нарушению предела памяти на 40 % тестов, если память выделяется динамически в зависимости от размера входных данных. Если же память выделяется по максимуму (30000 x 1000), не проходит ни один тест.

### Задача 4: Игра в 9999

Это – типичная задача, относящая к классу антагонистических игр. Похожая по концепции (но не по условию) задача была предложена на районной олимпиаде в 2006 году.

Её решение заключается в следующем. Разобьём все числа от 1 до 9999 на три категории:

- *финальное* число 9999;
- *выигрышные* числа, которые согласно правилам игры можно преобразовать хотя бы в одно проигрышное число;

- *проигрышные* числа, любое преобразование которых приводит к переходу в выигрышное число или в финальное число 9999.

При правильной стратегии игрок, обрабатывающий выигрышное число, должен преобразовать его в проигрышное. Если же очередное число – проигрышное, то игрок либо преобразует его в финальное (и проигрывает), либо отдаёт ход сопернику в выигрышной позиции.

Итак, ведущий выигрывает, если он придумал выигрышное число.

Опишем теперь, как определять категорию каждого числа. Очевидно, числа 8999, 9899, 9989, 9998 являются проигрышными. Организуем цикл от 9997 до 1 и для текущего числа (если его категория ещё неизвестна) пытаемся выполнить все возможные преобразования. После их выполнения категория числа определяется однозначно.

Рекомендуется вначале определить категорию каждого числа, а потом уже обрабатывать подтесты (т.е. выполнить т.н. *предпросчёт*).

# СОДЕРЖАНИЕ

<b>ПРЕДИСЛОВИЕ</b>	<b>3</b>
<b>УСЛОВИЯ ЗАДАЧ</b>	<b>5</b>
<b>2013/14 учебный год</b>	<b>5</b>
Задача 1: Испытания поезда	5
Задача 2: Место для фабрики	5
Задача 3: Набрать SMS	6
Задача 4: Дроби	8
<b>2014/15 учебный год</b>	<b>9</b>
Задача 1: The king and the knight	9
Задача 2: Слова-перевёртыши	10
Задача 3: Разрисованная плоскость	11
Задача 4: Подняться на лестницу	12
<b>2015/16 учебный год</b>	<b>14</b>
Задача 1: Районная олимпиада по информатике	14
Задача 2: Чекер для sudoku	16
Задача 3: Железнодорожная изгородь	18
Задача 4: Игра в 9999	19
<b>РАЗБОР ЗАДАЧ</b>	<b>21</b>
<b>2013/14 учебный год</b>	<b>21</b>
Задача 1: Испытания поезда	21
Задача 2: Место для фабрики	21
Задача 3: Набрать SMS	22
Задача 4: Дроби	22
<b>2014/15 учебный год</b>	<b>23</b>
Задача 1: The king and the knight	23
Задача 2: Слова-перевёртыши	23
Задача 3: Разрисованная плоскость	24
Задача 4: Подняться на лестницу	25
<b>2015/16 учебный год</b>	<b>26</b>
Задача 1: Районная олимпиада по информатике	26
Задача 2: Чекер для sudoku	26
Задача 3: Железнодорожная изгородь	27
Задача 4: Игра в 9999	27

Учебное издание

**Кашкевич Сергей Иванович  
Толстиков Алексей Александрович**

**СБОРНИК  
ОЛИМПИАДНЫХ ЗАДАЧ  
ПО ИНФОРМАТИКЕ**

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

**В пяти частях**

**Часть 1**

В авторской редакции

Ответственный за выпуск *А. А. Толстиков*

Подписано в печать 24.02.2016. Формат 60×84/16. Бумага офсетная.

Усл. печ. л. 1,63. Уч.-изд. л. 1,13. 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.