



*Кировское областное государственное автономное образовательное
учреждение дополнительного образования
«Центр дополнительного образования одаренных школьников»*

ИНФОРМАТИКА, ЧАСТЬ 2, 2015

ЗАДАНИЯ, РЕШЕНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по проведению
II (муниципального) этапа
всероссийской олимпиады школьников
по информатике

в Кировской области
в 2015/2016 учебном году

Киров
2015

Печатается по решению региональной предметно-методической комиссии всероссийской олимпиады школьников по информатике в Кировской области

Задания, решения и методические указания по проведению II (муниципального) этапа всероссийской олимпиады школьников по информатике в Кировской области в 2015/2016 учебном году. Часть 2 / Сост. Н.А. Будин, И.С. Кайсин, А.С. Латышев, О. А. Пестов // Под ред. О. А. Пестова. – Киров: Изд-во ЦДООШ, 2015. – 12 с.

Авторы, составители:
Будин Николай Алексеевич,
Кайсин Илья Сергеевич,
Латышев Алексей Сергеевич,
Пестов Олег Александрович

Научное рецензирование:
кандидат педагогических наук С. Ю. Иванов

Компьютерный набор О. Пестова
Подписано в печать 21.09.2015
Формат 60х84 1/16. Бумага типографская. Усл. печ. л. 0,85
Техническая редакция – ЦДООШ
Тираж 684 экз.

© Н. А. Будин, И. С. Кайсин, А. С. Латышев, О. А. Пестов, 2015

© Кировское областное автономное образовательное учреждение дополнительного образования «Центр дополнительного образования одарённых школьников», Киров, 2015

ОБЩИЕ ПРАВИЛА

1. Олимпиада проводится в компьютерной форме с использованием централизованной автоматической тестирующей системы. Формат проведения – практическое решение задач. Решения сдаются в тестирующую систему, доступ в которую осуществляется через сеть интернет со страницы: orlab.org/olymp43. Участники видят результат проверки во время основного тура.

2. Решением задачи является программа на одном из допустимых языков программирования, которые поддерживаются тестирующей системой. Это Pascal, C/C++, Python, Java, C#. Сдаваемая на проверку программа должна быть консольным приложением, не использующим какие-либо графические возможности операционной системы (диалоговые окна, формы ввода, средства рисования и т. д.). Программа должна читать данные со стандартного ввода (клавиатуры) и выводить результат на стандартный вывод (экран).

3. За решение одной задачи участник может набрать 100 баллов. Для этого программа должна вывести правильный ответ на всех заранее подготовленных тестах. Если программа работает не на всех тестах, то она получает частичный балл в зависимости от количества пройденных тестов.

4. Сдавать решение можно несколько раз. Количество посылок не влияет на итоговый результат. Из всех решений в зачет идет набравшее наибольшее число баллов.

5. Разрешается использовать литературу на бумажном носителе: книги, документация, личные записи. Запрещается использовать ресурсы сети интернет (кроме сайта тестирующей системы), средства связи, электронное оборудование (кроме рабочего компьютера) и носители информации.

6. Все вопросы по условиям задач участники задают через тестирующую систему (ссылка «Отправить вопрос»). Ответ можно увидеть в тестирующей системе на странице «Сообщения».

7. Материалы олимпиады (решения жюри и тесты) будут доступны на странице orlab.org/olymp43 вечером в день проведения олимпиады. В случае несогласия с результатом, участники имеют право подать обоснованную апелляцию.

ИНСТРУКЦИЯ ДЛЯ ЖЮРИ

1. Муниципальный этап всероссийской олимпиады школьников по информатике в Кировской области состоится 1 ноября 2015 года. Продолжительность основного тура — 3 астрономических часа. Время начала организаторы выбирают самостоятельно.

2. До олимпиады необходимо подготовить рабочие места участников. На компьютере должны стоять такие среды разработки, чтобы участник мог писать программы, используя следующие языки программирования: Pascal (PascalABC.Net, Free Pascal), C/C++ (Code::Blocks, Microsoft Visual C++ Express Edition), C# (MonoDevelop, Xamarin Studio, Microsoft Visual C# Express Edition), Java (JDK, Eclipse), Python (Python 2, Python 3, WingIDE 101, PyCharm). Не обязательно устанавливать все программы – достаточно только необходимые участникам.

3. Все компьютеры участников должны быть подключены к интернету. Рекомендуется использовать фильтрацию на уровне сервера, ограничивающую доступ только к сайту orlab.org/olymp43. При невозможности фильтрации доступа техническими средствами, во время тура должно осуществляться наблюдение за работой учащихся.

4. Перед проведением олимпиады, членам жюри настоятельно рекомендуется принять участие в пробном туре, который будет доступен на странице: orlab.org/olymp43 с 16 по 31 октября 2015 года. Это связано с тем, что проверка решений автоматической системой накладывает определенные требования на оформление программ. Эти требования нужно знать, чтобы донести их до участников. В частности необходимо объявить участникам, что они:

- не должны «разговаривать» с компьютером. Нельзя выводить приглашения перед чтением вида `writeln('Введите число:')` или возможные «обрамления» ответа вида `writeln('Ответ = ')`. Необходимо строго следовать формату входных и выходных данных;
- не должны проверять вводимые данные на корректность. Если написано, например, что число положительное и не больше ста, то гарантируется, что это так.

5. Перед началом олимпиады жюри получает условия задач и список учетных записей для входа в проверяющую систему. Каждому участнику олимпиады жюри назначает индивидуальный логин и пароль из полученного списка.

6. До начала тура участникам предоставляется возможность проверить работоспособность компьютера, сред разработки, логина в тестирующую систему.

7. Тур начинается после того, как устранены все технические проблемы. Началом тура является выдача условий участникам. Время начала тура записывается на доске.

8. В любых нештатных ситуациях, когда нет возможности сдать задания в тестирующую систему (например, пропало подключение к интернету) необходимо организовать сбор решений участников, архив решений сохраняется у организаторов и передается в предметно-методическую комиссию.

9. При наличии сбоев в работе тестирующей системы тур не продлевается, учащиеся продолжают решать задачи. Порядок действий аналогичен предыдущему пункту.

10. Несмотря на то, что для проверки олимпиады используется автоматическая тестирующая система, поддерживаемая предметно-методической комиссией, результаты олимпиады подводит жюри муниципального этапа. Жюри олимпиады должно после окончания тура для каждого участника записать его результат из тестирующей системы и составить протокол проведения олимпиады.

11. Победители и призёры муниципального этапа олимпиады определяются отдельно по классам.

12. Сразу после выполнения заданий проводится разбор решений, о чем следует объявить учащимся заранее, перед началом олимпиады.

13. В случае несогласия с результатом, участники имеют право подать обоснованную апелляцию. Такие ситуации решаются в индивидуальном порядке — для их разрешения необходимо связаться с предметно-методической комиссией.

14. Для связи с комиссией необходимо написать Пестову Олегу Александровичу oleg.pestov@gmail.com. Для экстренной связи можно звонить по телефону +7(909)143-57-74.

КОМПЛЕКТ ОЛИМПИАДНЫХ ЗАДАЧ

Для проведения муниципального этапа олимпиады по информатике предметно-методическая комиссия разработала комплект из шести задач.

В разных классах используются разные задачи:

- 7 «Торт», «Приветствие», «Боулинг», «Сортировка 78»;
- 8 «Торт», «Приветствие», «Боулинг», «Сортировка 78»;
- 9 «Торт», «Приветствие», «Сортировка 91011», «Переливания»;
- 10 «Торт», «Приветствие», «Сортировка 91011», «Переливания»;
- 11 «Торт», «Приветствие», «Сортировка 91011», «Переливания».

По мнению комиссии задачи в комплекте упорядочены в порядке увеличения сложности. Первые являются простыми и должны быть доступны практически всем участникам муниципального этапа. В свою очередь, последние задачи являются задачами повышенной сложности и ориентированы на сильнейших участников.

Важно! Во всех задачах кроме «Сортировка 78» и «Сортировка 91011», ограничение времени работы решения на одном тесте составляет одну секунду. В задачах «Сортировка 78» и «Сортировка 91011», ограничение времени работы решения на одном тесте составляет две секунды. Ограничение на объём используемой памяти во всех задачах 64 мегабайта.

ЗАДАЧИ

Задача «Торт»

Есть квадратный торт размера $N \times N$ клеток. Необходимо расставить на нем M свечек так, чтобы они стояли симметрично относительно вертикали и горизонтали. Напишите программу, которая определит можно ли это сделать.

Формат входных данных

В первой строке через пробел записаны целые числа N и M ($1 \leq N \leq 100$; $1 \leq M \leq 100000$).

Формат результата

Выведите Yes если свечи расставить можно и No, если нельзя.

Примеры

стандартный ввод	стандартный вывод
3 5	Yes
4 5	No

Примечание. Один из способов расставить свечи в первом примере:

.С.
ССС
.С.

Система оценки

В этой задаче 50 тестов. Тесты оцениваются независимо по два балла за тест.

Задача «Приветствие»

Напишите программу, которая читает имя и выводит приветствие Hello, <имя>!

Формат входных данных

В единственной строке находится непустая последовательность латинских букв. Длина имени не превышает ста.

Формат результата

Выведите три строки одинаковой длины. Первая строка начинается символом p и заканчивается символом q. Между ними несколько раз записан знак минус. Вторая строка содержит приветствие, по обе стороны которого находятся ровно один пробел и вертикальная черта. Третья строка начинается символом b и заканчивается символом d, между которыми несколько раз записан знак минус.

Примеры

стандартный ввод	стандартный вывод
Joe	p-----q Hello, Joe! b-----d

Система оценки

В этой задаче 25 тестов. Тесты оцениваются независимо по четыре балла за тест. В двух тестах имя состоит из четырех символов. В трех тестах имя состоит из пяти символов.

Задача «Боулинг»

Партия в игре в боулинг состоит из 10 туров. Задача игрока – в каждом туре сбить шаром как можно больше кеглей из 10 возможных, для чего ему предоставляется два броска шара. Если 10 кеглей сбиты первым броском, то второй бросок не совершается. Если 10 кеглей сбиты первым броском в десятом туре, то игроку предоставляются два призовых броска, а если двумя бросками, то – один.

Количество очков в каждом туре равно количеству сбитых кеглей, кроме двух бросков, называемых «Strike» и «Spare». При броске «Strike» игрок сбивает 10 кеглей первым броском, и очки в этом туре начисляются из расчета – (10 + сумма очков за два последующих броска). При броске «Spare» игрок сбивает 10 кеглей двумя бросками, очки в этом туре начисляются из расчета – (10 + сумма очков за один последующий бросок). Результат партии складывается из результатов всех 10 туров.

Требуется по количеству бросков и сбитых в каждом из них кеглей определить количество набранных игроком очков.

Формат входных данных

В первой строке записано количество бросков. Вторая строка содержит разделенные пробелом числа, обозначающие количество сбитых кеглей за соответствующий бросок.

Формат результата

Выведите количество набранных очков.

Примеры

стандартный ввод	стандартный вывод
12 10 10 10 10 10 10 10 10 10 10 10 10	300
15 10 10 10 8 2 10 3 4 8 2 4 5 10 4 5	173

Система оценки

Тесты к этой задаче состоят из трех групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы.

Группа	Тесты	Баллы	Комментарий
1	1-15	30	Нет бросков Strike и Spare
2	16-30	30	Нет бросков Strike
3	31-50	40	

Задача «Сортировка 78»

Дан массив целых чисел. Требуется упорядочить его элементы по неубыванию с помощью операций обмена соседних элементов. Напишите программу, которая для заданного массива длины n выведет последовательность обменов.

Формат входных данных

В первой строке задается целое число n ($1 \leq n \leq 500$). Во второй строке через пробел записаны n целых чисел $a_1, a_2, \dots, a_n$ ($-10000 \leq a_i \leq 10000$).

Формат результата

Выведите последовательность обменов по одному на строке. Каждый обмен задается парой индексов соседних элементов, которые меняются на соответствующем шаге. Количество обменов в ответе не должно превышать 125000. Если существует несколько последовательностей обменов, то выведите любую.

Примеры

стандартный ввод	стандартный вывод
1 3	
3 9 25 17	2 3
4 3 1 -1 -1	2 3 2 1 2 3 3 4 3 2

Система оценки

Тесты к этой задаче состоят из трех групп. Баллы за первые две группы ставятся только при прохождении всех тестов группы. Тесты в третьей группе оцениваются независимо по пять баллов за тест.

Группа	Тесты	Баллы	Комментарий
1	1-4	10	n = 2
2	5-15	15	n = 3
3	16-30	75	

Задача «Сортировка 91011»

Дан массив целых чисел. Требуется упорядочить его элементы по неубыванию с помощью *минимального* числа операций обмена соседних элементов. Напишите программу, которая для заданного массива длины n выведет последовательность обменов.

Формат входных данных

В первой строке задается целое число n ($1 \leq n \leq 100000$). Во второй строке через пробел записаны n целых чисел $a_1, a_2, \dots, a_n$ ($-10^9 \leq a_i \leq 10^9$). Гарантируется, что массив может быть упорядочен за не более, чем 125000 обменов.

Формат результата

Выведите последовательность обменов по одному на строке. Каждый обмен задается парой индексов соседних элементов, которые меняются на соответствующем шаге. Если существует несколько минимальных последовательностей обменов, то выведите любую.

Примеры

стандартный ввод	стандартный вывод
1 3	
3 9 25 17	2 3
4 3 1 -1 -1	2 3 2 1 2 3 3 4 3 2

Система оценки

Тесты к этой задаче состоят из четырех групп. Баллы за первые две группы ставятся только при прохождении всех тестов группы. Тесты в третьей группе оцениваются независимо по три балла за тест. Тесты в четвертой группе оцениваются независимо по пять баллов за тест.

Группа	Тесты	Баллы	Комментарий
1	1-4	5	$n = 2; -10^4 \leq a_i \leq 10^4$
2	5-15	10	$n = 3; -10^4 \leq a_i \leq 10^4$
3	16-30	45	$n \leq 500; -10^4 \leq a_i \leq 10^4$
4	31-38	40	

Задача «Переливания»

Дано три емкости вместимостью A , B и C литров. Изначально все емкости пусты. Также в распоряжении имеется источник воды. Разрешено делать одно из следующих действий:

- полностью наполнить одну из емкостей;
- полностью опустошить одну из емкостей;
- перелить воду из одной емкости в другую. Вода переливается полностью или переливание останавливается, когда емкость в которую вода наливается полна.

Требуется с помощью этих действий получить X литров в одной из емкостей или определить, что это невозможно.

Формат входных данных

В первой строке записаны четыре неотрицательных целых числа A , B , C и X . Все числа не превышают ста и $X \geq 1$.

Формат результата

Выведите состояния, получающиеся в процессе переливаний. Состояние выводится на отдельной строке и описывается тремя числами — количеством воды в первой, второй и третьей емкостях соответственно. Первым должно быть состояние 0 0 0. В последнем состоянии в одной из емкостей должно быть X литров. Количество состояний в ответе не должно превышать 10000. Если искомой последовательности переливаний не существует, то выведите 0 0 0.

Примеры

стандартный ввод	стандартный вывод
5 8 2 4	0 0 0 0 0 2 0 2 0 0 2 2 0 4 0
4 8 0 2	0 0 0
5 8 0 4	0 0 0 5 0 0 0 5 0 5 5 0 2 8 0 2 0 0 0 2 0 5 2 0 0 7 0 5 7 0 4 8 0

Система оценки

Тесты к этой задаче состоят из трех групп. Баллы за первую группу ставятся только при прохождении всех тестов группы. Тесты во второй и третьей группах оцениваются независимо по четыре балла за тест.

Группа	Тесты	Баллы	Комментарий
1	1-5	20	$B = 0, C = 0$
2	6-15	40	$C = 0$
3	16-25	40	

УКАЗАНИЯ ПО РЕШЕНИЮ ЗАДАЧ

Задача «Торт»

Если N – четное, то при постановке одной свечи необходимо поставить еще три, чтобы выполнялось условие симметричности. В этом случае ответ Yes, только если $M \leq N^2$ и M делится нацело на четыре.

Если N – нечетное, то свечи можно расставить всегда. Конечно при условии, что их не слишком много, то есть если $M \leq N^2$. Доказательство существования решения может быть конструктивным. При нечетном N рассмотрим отдельно клетки в центральной строке и центральном столбце — их $2N-1$. На оставшуюся часть торта свечи ставятся по четыре из-за условия симметричности. Итак, до тех пор пока количество свечей больше $2N-1$ берем четыре и ставим их симметрично в часть отличную от центральной. После этого пока остается более одной свечи, берем две и ставим их симметрично в свободное место в центральной строке или центральном столбце. Если в конце осталась одна свеча, то ставим ее в центр.

Задача «Приветствие»

Необходимо прочитать строку, узнать ее длину и в цикле создать строку из минусов длина которой на 10 больше, чем длина имени.

Также можно не создавать вспомогательную строку, а сразу выводить ответ.

```
readln(s);
s := 'Hello, ' + s + '!';
write('p-'); for i := 1 to length(s) do write('-'); writeln('-q');
write('| '); write(s); writeln(' |');
write('b-'); for i := 1 to length(s) do write('-'); writeln('-d');
```

Задача «Боулинг»

Один из вариантов решения — это для каждого из 10 раундов узнать номер первого броска в этом раунде. Пусть это будет вспомогательный массив R . Например, если $R[3]=4$, то четвертый бросок был первым броском в третьем раунде. Массив можно заполнить в цикле.

```
r[1] := 1;
for i := 2 to 10 do
  if a[r[i - 1]] = 10 then
    r[i] := r[i - 1] + 1
  else
    r[i] := r[i - 1] + 2;
```

С помощью R можно найти количество очков в любом раунде i :

- если $A[R[i]]=10$, то это Strike и набрано $10+A[R[i]+1]+A[R[i]+2]$ очков;
- иначе если $A[R[i]]+A[R[i]+1]=10$, то это Spare и набрано $10+A[R[i]+2]$ очков;
- иначе очки это $A[R[i]]+A[R[i]+1]$

Задача «Сортировка 78»

Заметим, что массив упорядочен по неубыванию, если в нем нет двух соседних элементов таких, что первый больше второго. Поэтому пока есть такие пары их нужно менять. Например, пробежаться по массиву и если есть два соседних элемента, нарушающих порядок, то поменять их местами.

```
for j := 1 to n - 1 do
  if a[j] > a[j + 1] then begin
    writeln(j, ' ', j + 1);
    x := a[j]; a[j] := a[j + 1]; a[j + 1] := x;
  end;
```

Одного прохода может быть недостаточно. Так, после одного такого прохода массив 4 3 1 2 превратится в 3 1 2 4. Но верно, что после первого прохода максимальный элемент встанет на последнее место. После второго прохода следующий после максимального встанет на предпоследнее место и т.д. То есть за $n-1$ проход массив точно будет упорядочен.

```
for i := 1 to n - 1 do
  for j := 1 to n - 1 do
    if a[j] > a[j + 1] then begin
      writeln(j, ' ', j + 1);
      x := a[j]; a[j] := a[j + 1]; a[j + 1] := x;
    end;
```

Такой алгоритм сортировки называется «сортировка пузырьком». Представленную реализацию можно улучшить и не переходить к следующей итерации в цикле по i , если на предыдущем шаге не было обменов. Но даже и без этой оптимизации сложность решения — $O(n^2)$. Что легко укладывается в ограничения по времени при $n = 500$. Количество обменов в худшем случае составляет $n(n-1)/2$ для перевернутого массива. При $n=500$ это число не превышает 125000.

Задача «Сортировка 91011»

Отличие данной задачи от задачи «Сортировка 78» состоит в большей длине массива, в требовании минимального числа обменов и при этом в гарантии, что 125000 обменов будет достаточно.

Введем понятие инверсии — это пара элементов (x, y) таких, что x больше, чем y , но x встречается в массиве раньше, чем y . В упорядоченном по неубыванию массиве инверсий нет. Если массив попарно различных элементов длины n упорядочен по убыванию, то в нем $n(n-1)/2$ инверсий. Когда меняются два соседних элемента, то убирается ровно одна инверсия, поэтому если сортировать массив с помощью обмена соседних элементов, то количество обменов *всегда* равно числу инверсий в исходном массиве. Это показывает, что требование минимальности в условии этой задачи не играет никакой роли.

Рассмотрим решение для задачи «Сортировка 78»:

```
for i := 1 to n - 1 do
  for j := 1 to n - 1 do
    if a[j] > a[j + 1] then begin
      writeln(j, ' ', j + 1);
      x := a[j]; a[j] := a[j + 1]; a[j + 1] := x;
    end;
```

Оно будет выводить правильный ответ, но при больших n будет работать слишком долго. Оптимальным здесь будет использовать сортировку вставками.

```
for i := 1 to n do begin
  j := i;
  while (j > 1) and (a[j] < a[j - 1]) do begin
    writeln(j - 1, ' ', j);
    x := a[j]; a[j] := a[j - 1]; a[j - 1] := x;
    j := j - 1;
  end;
end;
```

На каждом шаге массив $a_1, a_2, \dots, a_{i-1}$ упорядочен и добавляется элемент a_i . До тех пор, пока он участвует в инверсиях — выполняются обмены. Сложность этого алгоритма — количество инверсий (в условии задачи гарантируется, что это число не превышает 125000).

Задача «Переливания»

Поскольку вместимость емкостей не превышает 100 литров, то количество возможных различных состояний не превышает $ABC=10^6$. В решении нужно перебрать все состояния достижимые из состояния 000. Если достижимо какое-то из состояний вида $X??$, $?X?$ или $??X$, то решение существует.

Перебор можно осуществлять с помощью обхода в ширину. На каждом шаге мы рассматриваем достижимое, но еще не рассмотренное состояние и пытаемся выполнить все доступные действия. В результате этих действий мы получим новые состояния. Если про эти состояния мы уже знаем, что они достижимы, то игнорируем их, иначе добавляем в список достижимых, но не рассмотренных.

Таким образом за $O(ABC)$ шагов мы обойдем все состояния.

Чтобы вывести последовательность состояний необходимо для каждого состояния хранить состояние, из которого оно было получено.

Детали реализации можно посмотреть в решениях жюри.