

**Задания муниципального этапа
всероссийской олимпиады школьников по информатике
2015-2016 учебный год**

7-8 классы

Время выполнения – 5 часов

Максимальное количество баллов – 400

Максимальное количество баллов за решение одной задачи – 100

Особенности проверки

На проверку принимаются тексты программ! Проверяющий компилирует текст программы и запускает исполняемый файл для каждого теста. Результат работы программы сравнивается с эталонным. Ответ должен полностью совпадать! В том числе по формату и порядку вывода чисел.

- Для всех задач имеются папки с файлами. Там содержится
- а) несколько тестовых примеров (как в формулировке задачи)
 - б) по 10 тестов (входные данные и эталонный ответ)
 - в) решения жюри.

На проверку принимаются только те решения, которые выдают правильные ответы для тестов, приведенных в примерах. **Баллы за прохождение таких тестов не начисляются!**

За прохождение каждого теста из основной группы начисляется 10 баллов.

При проверке задач необходимо соблюдать следующие требования:

- а) хотя бы визуально контролировать время выполнения, оно не должно превышать 1 секунду, а программа не должна зависать;
- б) проверять текст программы визуально, при этом программы-пустышки, выводящие одинаковый результат независимо от входных данных, выдающие конкретное число на конкретные входные данные (то есть не содержащие как такового алгоритма), оцениваются в 0 баллов!

Просим также обратить внимание, что проверка решений участников на региональном этапе будет выполняться на freepascal, там, в отличие от ABC Pascal, тип

integer является 2-байтовым знаковым и имеет ограничение $-32768..32767$

longint является 4-байтовым знаковым и имеет ограничение $-2^{31}..2^{31}-1$

Ниже в тексте после каждой задачи приведены краткие объяснения по ее решению. Для лучшего понимания можно посмотреть решения жюри.

Первые три задачи не требуют файлового ввода-вывода. В четвертой задаче использование файлов обязательно.

Задача 1. Пифагорова тройка

Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	64 мегабайта
Максимальная оценка:	100 баллов
Входной файл	pf_in.txt
Выходной файл	pf_out.txt

В математике известны тройки чисел, которые называются «Пифагоровы». Это такие целые числа **A**, **B** и **C**, для которых выполняется соотношение: $A^2 + B^2 = C^2$. Можно ли для каждого числа **C** найти такие числа **A** и **B**, чтобы **A**, **B** и **C** образовывали Пифагорову тройку?

Формат входных данных (допускается ввод с клавиатуры)

В первой строке через пробел вводится одно целое число **C** – последнее число из тройки Пифагора. $2 \leq C \leq 30\,000$.

Формат выходных данных (допускается вывод на экран)

В одной строке выведите через пробел два целых числа **A** и **B**, образующих вместе с числом **C** тройку Пифагора: $A^2 + B^2 = C^2$. Если для заданного **C** таких чисел не может быть найдено, выведите -1.

Система оценивания

Максимум – 100 баллов.

Баллы начисляются за каждый пройденный тест.

Примеры входных и выходных данных

№ теста	Входные данные	Выходные данные
1	5	2 3
2	6	-1

КОММЕНТАРИИ К РЕШЕНИЮ

Данная задача решается полным перебором всех возможных **A**. Опишем основные подходы при переборе.

1) **A** перебирается от 1 до **C**-1.

2) При переборе для некоторого **A** всегда можно найти предполагаемое $Y = B^2 = C^2 - A^2$

3) Остается проверить, извлекается ли целый корень из **Y**. Например, можно так. **B** = округлить (корень ($C^2 - A^2$)). Если $A^2 + B^2 = C^2$, то решение найдено.

Участники должны выбрать правильный тип данных, если значение $C^2 - A^2$ присваивается некоторой переменной, ведь квадрат величины может превышать тип `integer`.

Задача 2. Наполни бочку

Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	64 мегабайта
Максимальная оценка:	100 баллов
Входной файл	barrel_in.txt
Выходной файл	barrel_out.txt

Имеется колодец и два ведра, емкостью **5** и **3** литра. С помощью этих ведер нужно налить в бочку ровно **C** литров, причем сделать это за минимальное число операций (наливаний, выливаний, переливаний), при этом брать воду из бочки нельзя. После завершения всех операций ведра должны быть пустыми, то есть при необходимости оставшуюся воду нужно вылить обратно в колодец.

Например, если $C = 1$, то количество операций 6:

1. Заполняем из колодца 3-литровое ведро.
2. Переливаем содержимое 3-литрового ведра в 5-литровое ведро.
3. Заполняем из колодца 3-литровое ведро.
4. Доливаем 5-литровое ведро до краев из 3-литрового ведра (остался 1 литр).
5. Оставшуюся воду из 3-литрового ведра заливаем в бочку.
6. Выливаем воду из 5-литрового ведра обратно в колодец (опорожняем ведро).

Требуется написать программу, которая выводит на экран минимальное число операций, необходимых для заполнения бочки ровно **C** литрами.

Формат входных данных (допускается ввод с клавиатуры)

В первой строке вводится одно целое **C** – сколько литров необходимо налить в бочку. $1 \leq C \leq 100$.

Формат выходных данных (допускается вывод на экран)

Выведите минимальное число операций, необходимых для заполнения бочки ровно **C** литрами.

Система оценивания

Максимум – 100 баллов.

Баллы начисляются за каждый пройденный тест.

Примеры входных и выходных данных

№ теста	Входные данные	Выходные данные
1	1	6
2	10	4

КОММЕНТАРИИ К РЕШЕНИЮ

Задача требует некоторой сообразительности. Выход за пределы допустимых типов не предполагается.

1 вариант решения.

Этот вариант может возникнуть после подробного изучения примера. Очевидно, что

1) Для получения C литров, кратных 5 необходимо сделать $(C \operatorname{div} 5) * 2$ операций;

2) В случае некрatности 5 можно получить остаток 1, 2, 3 или 4. Тогда можно вручную сосчитать, что для получения 2 литров нужно 4 операции, для получения 3 литров – 2 операции, а для получения 4 литров – 8 операций. Тогда можно рассчитать общее число операций по формуле:

$(C \operatorname{div} 5) * 2 + O$, где O берем в соответствии с остатком.

3) Аналогичные выкладки можно получить при рассуждениях о кратности 3.

4) Осталось рассмотреть случай, когда число C можно представить в виде $C = 5 * x + 3 * y$. Числа x , y можно найти перебором. Тогда число операций будет $= (x + y) * 2$.

5) Из трех полученных ответов теперь нужно выбрать минимальное.

Этот вариант не оптимальный и достаточно трудоемкий, хотя он полностью решает задачу и не имеет проблем с оценкой по времени выполнения, так как все числа достаточно малы.

2 вариант решения.

Из п. 4) можно догадаться, что правильное решение можно найти из целочисленного решения уравнения $C = \pm x * 5 \pm y * 3$. При этом число операций вычисляется как $2 * (|x| + |y|)$.

Тогда программно достаточно построить двойной вложенный цикл для перебора всех x и y от -100 до 100. Следует также помнить, что нужно найти минимальное число операций, поэтому в процессе поиска выбирать минимальное подходящее решение.

Задача 3. Подстрока в числе

Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	64 мегабайта
Максимальная оценка:	100 баллов
Входной файл	subs_in.txt
Выходной файл	subs_out.txt

В любом числе можно выделить некоторую непрерывную последовательность цифр, которая тоже будет некоторым числом. Требуется написать программу, которая находит максимальное натуральное число X такое, что десятичная запись числа X^2 является подстрокой в десятичной записи числа N . Если такого числа нет, то вывести ноль.

Формат входных данных (допускается чтение с клавиатуры)

Вводится натуральное число N . $1 \leq N \leq 1\,000\,000\,000$.

Формат выходных данных (допускается вывод на экран)

Выведите максимальное натуральное число X такое, что десятичная запись числа X^2 является подстрокой в десятичной записи числа N .

Система оценивания

Максимум – 100 баллов.

Баллы начисляются за каждый пройденный тест.

Примеры входных и выходных данных

№ теста	Входные данные	Выходные данные
1	21	1
2	333	0
3	9646251	25

КОММЕНТАРИИ К РЕШЕНИЮ

Данная задача элементарно решается перебором по возможным значениям X от корня из N до 1 с использованием строк и соответствующих функций для их обработки: `str` и `pos`. Данное решение приведено.

Однако, участники, не умеющие работать со строками, могут решить задачу с использованием массивов. Хотя, такой вариант требует хорошей техники программирования и является достаточно трудоемким. Это решение тоже приведено жюри. Сложность такого алгоритма линейная, поэтому ограничения по времени не будут превышены.

Задачу можно считать не сложной для понимания, но требующей навыков программирования.

Задача 4. Букеты

Максимальное время работы на одном тесте:	1 секунда
Максимальный объем используемой памяти:	64 мегабайта
Максимальная оценка:	100 баллов
Входной файл	flower_in.txt
Выходной файл	flower_out.txt

На 1 сентября ученики школы № 555 решили подарить каждому учителю по букету цветов. Чтобы никого не обидеть, было решено, что в каждом букете будет обязательно одинаковое нечетное число цветов одного цвета (окраса). Цветы привезли в школу в N коробках, в каждой из коробок лежат цветы одного и того же своего (уникального) окраса (например, в первой коробке – только красные, во второй – синие, в третьей розовые и т.д.)

Зная, что в школе работают K учителей, требуется написать программу, определяющую максимальное число цветов в каждом букете.

Формат входных данных

В первой строке входного файла содержатся разделенные пробелом натуральные числа K и N ($1 \leq K, N \leq 1000$). В следующих N строках содержится по одному натуральному числу A_i ($1 \leq A_i \leq 1\,000\,000$; $1 \leq i \leq N$), обозначающему количество цветов в i -й коробке.

Формат выходных данных

В выходной файл необходимо вывести одно целое число – количество цветов в каждом букете. Если для указанных условий создать букеты не удастся, необходимо вывести 0.

Система оценивания

Максимум – 100 баллов.

Баллы начисляются за каждый пройденный тест.

Примеры входных и выходных данных

№ теста	flow_in.txt	flow_out.txt
1	2 3 1 1 2	1
2	2 3 500 30 100	249
3	5 2 100 120	39
4	5 2 1 3	0

КОММЕНТАРИИ К РЕШЕНИЮ

Проверяются **ТОЛЬКО** решения с файловым вводом-выводом.

Данная задача считается сложной и предполагает невысокий процент решения.

Отметим следующие моменты.

- 1) Число цветов в букете обязательно нечетно!
- 2) Если есть некоторое предполагаемое число цветов в букете – F , то можно определить число букетов, которые в принципе можно создать из имеющихся наборов цветов: $K_f = \sum_{1 \leq i \leq N} (A_i \text{ div } F)$.

1 вариант решения.

Перебор всех вариантов F от $\max_{1 \leq i \leq N} (A_i)$ до 0 с учетом нечетности. Если K_f для некоторого F получается больше, чем требуемое K , то можно попробовать его увеличить. Продолжая перебор, найдем максимальное подходящее F .

Следует учесть, что мы получаем два вложенных цикла: первый для перебора F и второй – для подсчета K_f . Теоретически, максимально возможное число операций = $(1\,000\,000 \text{ div } 2) * 1000$. Такой перебор не укладывается в 1 сек. Но с учетом специфики задачи можно догадаться, что в большинстве тестов не будет таких крайних значений и ответ найдется за приемлемое время.

Данное решение приведено жюри.

2 вариант решения.

С учетом предыдущих выкладок задача решается двоичным поиском по ответу.

- 1) Пусть $L_f = 0$, $R_f = 1\,000\,000$. Левая и правая границы поиска.
 - 2) Возьмем за F среднее значение = $(L_f + R_f) / 2$
 - 3) Проверим, насколько подходит F , вычисляя K_f .
 - 4) Если $K_f < k$, то можно откинуть левую часть и взять $L_f = (L_f + R_f)/2$, иначе, откидывает правую часть $R_f = (L_f + R_f)/2$ и продолжаем поиск с шага 2).
 - 5) Процесс останавливается, когда границы будут на расстоянии 1, останется выбрать правильный ответ.
 - 6) Необходимо помнить, что ответ – нечетное число.
- Данное решение тоже приведено жюри.