



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

ИНФОРМАТИКА, 2012

**ЗАДАНИЯ, РЕШЕНИЯ
И МЕТОДИЧЕСКИЕ УКАЗАНИЯ**
по проверке и оценке решений
II (муниципального) этапа
Всероссийской олимпиады школьников
по информатике
в Кировской области
в 2012/2013 учебном году

**Киров
2012**

Печатается по решению методической комиссии II (муниципального) этапа Всероссийской олимпиады школьников по информатике в Кировской области

Задания, решения и методические указания по проверке и оценке решений II (муниципального) этапа Всероссийской олимпиады школьников по информатике в Кировской области в 2012/2013 учебном году / Сост. Р.А. Веснин, С. Ю. Иванов, О. А. Пестов // Под ред. С. Ю. Иванова, О. А. Пестова. – Киров: Изд-во ЦДООШ, 2012. – 12 с.

Авторы, составители:
Веснин Роман Александрович,
Иванов Сергей Юрьевич,
Пестов Олег Александрович

Научная редакция (рецензирование):
С. Ю. Иванов, О. А. Пестов

Компьютерный набор и верстка: С. Иванов, О. Пестов
Техническая редакция: К. Коханов
Подписано в печать 05.10.2012
Формат 60х84 1/16. Бумага типографская. Усл. печ. л. 0,75

Тираж 1000 экз

© Р.А. Веснин, С. Ю. Иванов, О. А. Пестов, 2012

© Кировское областное автономное образовательное учреждение дополнительного образования детей – «Центр дополнительного образования одаренных школьников», Киров, 2012

ОБЩИЕ РЕКОМЕНДАЦИИ ОРГКОМИТЕТУ И ЖЮРИ МУНИЦИПАЛЬНОГО ЭТАПА ВСЕРОССИЙСКОЙ ОЛИМПИАДЫ ШКОЛЬНИКОВ ПО ИНФОРМАТИКЕ

1. В данном комплекте представлены следующие материалы:

- тексты задач;
- вопросы для тестового задания;
- общие методические рекомендации по проверке решений участников;
- система оценки результатов прохождения каждого теста;
- краткие методические рекомендации по решению предложенных задач.

В связи проверкой решений участников при помощи онлайн-системы в состав данного комплекта не включены наборы тестов в электронном виде, содержащие для каждой задачи файлы входных данных и соответствующие им файлы выходных данных. При необходимости тесты можно получить, обратившись по адресу: olymp43@gmail.com.

2. Перед проведением олимпиады рекомендуется внимательно ознакомиться с предлагаемыми задачами и подготовить системы для тестирования решений. *Материалы должны быть переданы жюри как минимум за 10 дней до начала тура*, чтобы жюри смогло в полной мере подготовиться к проведению олимпиады. В случае возникновения вопросов по использованию автоматизированной системы проверки решений, формулировкам задач и тестам можно обращаться по e-mail: olymp43@gmail.com.

3. Длительность тура не должна превышать пяти астрономических часов, может быть организован пробный тур – знакомство участников с компьютерной техникой перед началом олимпиады.

4. Во время олимпиады участникам разрешается задавать вопросы по формулировкам задач, если в них содержится информация, которая может быть истолкована по-разному. Желательно, чтобы участник формулировал свой вопрос так, чтобы на него можно было ответить: «да», «нет» или «без комментариев».

5. Для решения задач участникам запрещается пользоваться литературой и различными электронными источниками.

6. Для решения задач участникам рекомендуется предоставить следующие среды программирования:

- PascalABC.Net
- Borland Delphi 7.0;
- FreePascal 2.4.0;
- CodeBlocks 10.05, Microsoft Visual Studio 2005.

Можно использовать и дополнительное инструментальное программное обеспечение.

Следует отметить, что на все программное обеспечение, используемое при проведении олимпиады, организаторы должны иметь необходимые лицензии.

Большинство рекомендуемых программных систем являются свободно распространяемыми и их можно загрузить с соответствующих сайтов. Примерами таких сайтов являются:

- PascalABC.Net – сайт <http://pascalabc.net>;
- FreePascal – сайт <http://freepascal.org>;
- MinGW – сайт <http://mingw.org>;
- Eclipse – сайт <http://eclipse.org>;
- Code::Blocks – сайт <http://www.codeblocks.org>.

В связи с тем, что по вопросу получения лицензионных прав на бесплатное использование продуктов Borland/Embarcadero на период олимпиады необходимо обращаться непосредственно в компанию Embarcadero Technologies, а также учитывая высокую стоимость продукта для персонального использования, в качестве альтернативы рекомендуется переход на PascalABC.Net или FreePascal.

7. Решением задачи должен являться один единственный файл на одном из разрешенных языков программирования.

8. Тестирование решения должно осуществляться следующим образом:

- компилирование файла с решением. Если файл не компилируется, то результатом является «ошибка компиляции» и за задачу не начисляется ни одного балла;
- запуск скомпилированного решения на одном из тестов. Если результат выведен корректно и совпадает с правильным ответом, то за тест начисляется соответствующее количество баллов. Иначе за тест не начисляется ни одного балла, и он заканчивается одной из следующих ошибок:

- «неверный формат вывода». Если решение участника ничего не выводит в качестве результата или выводит в неправильном формате;
- «неправильный ответ». Если результат выведен корректно, но не совпадает с правильным ответом;
- «ошибка времени исполнения». Если во время работы решения возникла необработанная ошибка;
- если решение на одном из тестов работает дольше отведенного времени для данной задачи, то тест завершается ошибкой «превышение лимита времени»;
- если решение на одном из тестов требует выделения объема памяти большего, чем отведено для данной задачи, то тест завершается ошибкой «превышение лимита памяти».

9. Тестирование можно осуществлять либо вручную, либо при помощи специальной автоматизированной системы проверки. Можно использовать для проверки решений участников одну из свободно распространяемых программных систем проведения олимпиад по информатике.

Тестовые задания не могут быть проверены автоматизированной системой, поэтому за их проверку и оценку отвечает жюри, ответственное за организацию этапа олимпиады. Ответы на тестовые задания приведены в разделе "Решения задач".

10. В сети Интернет находится автоматизированная система проверки решений по адресу: <http://olymp43.dyndns.org/>. Данная система работает под управлением операционной системы Linux, как следствие все доступные компиляторы также являются версиями для Linux:

- PascalABC.NET 1.8;
- Borland Kylix 3.0 (вместо Borland Delphi 7.0);
- FreePascal 2.4.0;
- GNU C/C++ 4.4.5;
- Mono C# 2.10.9;
- Java 1.7;
- Python 3.1.3;
- Perl 5.10.1;
- PHP 5.3.5.

До начала олимпиады на этом сайте доступен пробный тур с задачами прошлого года. Цель пробного тура – предоставить возможность членам жюри опробовать работу с системой до проведения олимпиады. Более того, вы можете провести тренировку для школьников: выдать им условия задач прошлого года, попросить написать решения и затем проверить в системе. Подробные инструкции по работе с системой доступны на сайте.

Порядок работы с системой до олимпиады

До 16 ноября 2012 г. по адресу: <http://olymp43.dyndns.org/> будет доступен пробный тур, его можно использовать для ознакомления работы с системой, а также для подготовки участников к олимпиаде. Результаты никак не будут учитываться при проведении самой олимпиады. Более того, имена входа и пароли для олимпиады также будут другими. Более подробные инструкции можно прочитать по адресу: <http://olymp43.dyndns.org/doc/guide.html>.

Порядок работы с системой во время олимпиады

На время олимпиады работа системы будет приостановлена, по окончании олимпиады жюри собирает решения участников, регистрирует всех участников в системе, заполняя следующую информацию: фамилия, имя, отчество ученика, его класс и школа, после чего производит вход в систему под определенным учеником и производит тестирование его решений. Более подробные инструкции доступны по адресу: <http://olymp43.dyndns.org/doc/guide.html>.

Обращаем внимание, что при использовании данной системы ввод и вывод осуществляется не с файлами, а со стандартными устройствами ввода/вывода (клавиатура и экран соответственно).

11. В связи с тем, что за прохождение каждого теста начисляется определенное количество баллов, то рекомендуется разъяснить участникам, что в случае отсутствия у них эффективного решения (которое бы полностью удовлетворяло всем ограничениям) имеет смысл писать «частичные» решения (например, корректно работающие при небольших входных данных).

12. Подведение итогов муниципального этапа предметно-методическая комиссия рекомендует осуществлять отдельно по классам.

УСЛОВИЯ ВСЕХ ЗАДАЧ

Ограничение по времени на все задачи: 1 секунда.

Ограничение по памяти на все задачи: 64 Мбайт.

Входной файл: стандартный ввод (с клавиатуры).

Выходной файл: стандартный вывод (на экран).

Максимальный балл за полное решение любой задачи составляет 100 баллов.

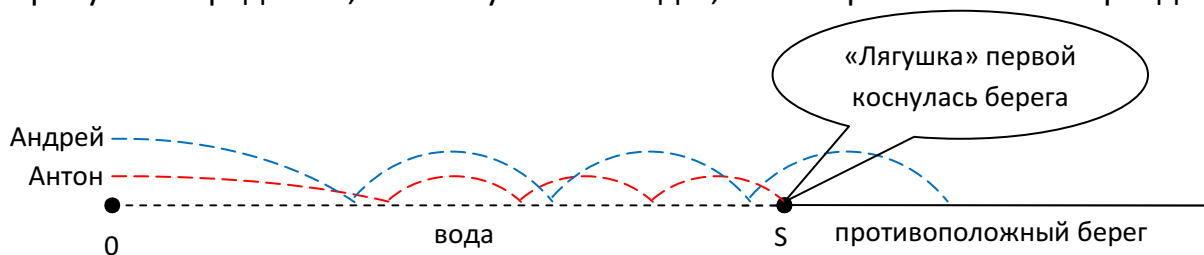
Тестовые задания также оцениваются исходя из 100 баллов, при этом за каждый правильный ответ начисляется по 10 баллов. Таким образом, максимальная сумма баллов, которую может набрать участник составляет 500 баллов.

1. «Лягушка»

Антон и Андрей кидали на озере камни «лягушкой» так, чтобы, коснувшись поверхности воды, камень отскакивал и продолжал «прыгать» дальше, пока не достигнет другого берега. Антон кидает камень на M метров, и его «лягушка» делает прыжки длиной X метров после первого касания воды. А Андрей кидает камень на N метров, и его «лягушка» продолжала прыгать по Y метров.

Они решили посоревноваться, чья «лягушка» первой коснётся дальнего берега, расстояние до которого составляло S метров. Антон и Андрей условились запускать «лягушек» одновременно с одной линии так, чтобы камни летели параллельно друг другу, их пути не пересекались и скорость была бы одинаковой.

Требуется определить, чья «лягушка» победит, и какое расстояние она преодолеет.



В данном примере победил Антон так, как его лягушка приземлилась ближе к берегу, а, следовательно, и раньше.

Входные данные. В первой строке содержатся пять целых чисел, разделённых пробелом: S, M, X, N, Y ($0 < S < 10^9$; $0 < M, N < S$; $0 < X, Y < 10^9$).

Выходные данные. В выходной файл выведите 1, если победил Антон, 2 – если победил Андрей и 0 если обе «лягушки» приземлились одновременно.

Если допустимых маршрутов несколько, выведите любой из них.

Пример

Входные данные	Выходные данные	Пояснения
10 9 1 2 6	1	Первая «лягушка» приземлится ровно на край другого берега ($9\text{м} + 1\text{м}$), вторая лягушка приземлится на расстоянии 4м от края берега ($2\text{м} + 6\text{м} + 6\text{м}$), следовательно, земли коснётся раньше первая «лягушка»
10 9 9 2 3	2	Первая «лягушка» приземлится на расстоянии 8м от края берега ($9\text{м} + 9\text{м}$), вторая лягушка приземлится на расстоянии 1м от края берега ($2+3+3+3$), следовательно, земли коснётся раньше вторая «лягушка»

2. По росту становись!

В школе все ребята выстроились в колонну друг за другом по росту (по невозрастанию), но лишь один ученик с меньшим ростом занял не своё место и встал перед более рослыми товарищами. Найдите ученика, стоящего не на своём месте.

Входные данные. В первой строке записано целое число N ($2 < N < 1000$) – количество учеников, далее идут N чисел (рост i -того ученика) в том порядке, в каком они выстроились в колонну. Рост каждого ученика лежит в диапазоне от 100 до 200.

Выходные данные. Выведите одно единственное число – номер ученика стоящего не на своём месте.

Пример

Входные данные	Выходные данные
5 150 125 130 125 120	2
8 120 150 149 145 143 142 140 115	1

3. Частота букв

Дан набор из N слов, каждое слово состоит только из маленьких латинских букв. Требуется определить, есть ли такие буквы латинского алфавита, которые встречаются во всех имеющихся словах.

Входные данные. В первой строке записано целое число N ($0 < N < 10^5$). Далее идут N слов, каждое слово записано на отдельной строке. Длина каждого слова не превышает 50 символов.

Выходные данные. Выведите в первую строку “YES” (без кавычек), если есть буквы, встречающиеся во всех словах, иначе выведите “NO” (без кавычек). Если в первой строке выведено “YES”, то во вторую строку выведите все буквы, которые встречаются во всех словах (в алфавитном порядке без пробелов)

Пример

Входные данные	Выходные данные	Пояснения
3 pizza lemon simple	NO	Ни одна из букв латинского алфавита не встречается во всех словах
5 rar rapid car bar arc	YES ar	Две буквы «а» и «г» встречаются во всех словах

4. Дроби

Рассмотрим способ нумерации дробей. Для этого поместим все дроби в таблицу.

1/1	1/2	1/3	1/4	...
2/1	2/2	2/3	2/4	...
3/1	3/2	3/3	3/4	...
4/1	4/2	4/3	4/4	...
...	...	...	...	...

В строку с номером i этой таблицы по порядку записаны дроби с числителем i , а в столбец с номером j , дроби со знаменателем j . Теперь выпишем дроби «змейкой», начиная из верхнего левого угла: 1/1 -> 2/1 -> 1/2 -> 1/3 -> 2/2 -> 3/1 -> 4/1 -> 3/2 -> 2/3 -> 1/4 -> 1/5 -> 2/4 -> 3/3 -> 4/2 ->...

Требуется найти N -ную дробь в последовательности.

Входные данные. В первой строке находится одно единственное целое число N ($1 \leq N \leq 10^{18}$).

Выходные данные. Требуется вывести через символ «/» два числа: числитель и знаменатель соответствующей дроби.

Пример

Входные данные	Выходные данные
2	2 / 1
10	1 / 4

Примечание. Частичные решения для $N \leq 100$ будут оцениваться в 40 баллов, для $N \leq 1000$ – в 60 баллов, для $N \leq 10^9$ – в 80 баллов.

5. Кратные числа

Выпишем на листе бумаги последовательно все числа от 1 до N . Далее выполним с ними K раз такую операцию: на очередном шаге выберем число a_i и обведём все числа кратные a_i . Если число уже было обведено, то обведём ещё раз.

Требуется сказать, сколько чисел мы не обвели ни разу.

Входные данные. В первой строке находятся два числа: N – количество чисел ($1 \leq N \leq 300000$) и K – количество шагов ($1 \leq K \leq 100000$). Во второй строке даны K целых чисел: $a_1, a_2, \dots, a_K$ ($2 \leq a_i \leq N$) – числа которые выбирались на каждом шаге.

Выходные данные. Выведите одно число – количество не обведённых чисел.

Пример

Входные данные	Выходные данные	Пояснения
5 3 4 2 4	3	Мы выписали числа от 1 до 5 и выполнили три шага. На первом обвели все числа кратные четырём – то есть 4. На втором шаге все числа кратные двум – 2, 4. На третьем опять 4. Мы ни разу не обвели три числа: 1, 3, 5. Ответ: 3.
8 2 2 3	3	На первом шаге мы обвели числа 2, 4, 6, 8. На втором – 3, 6. Ни разу не обвели 1, 5, 7. Ответ: 3.

6. Тестовые задания для учащихся из 7–8-х классов

1) Выберите варианты, в которых указан объём информации больше 2-х килобайт: 2012 байт, 2021 байт, 2050 байт, 2100 байт, 2^{14} бит.

2) Модем, передающий информацию со скоростью 128 Кбит/с, передал текстовый файл, содержащий 65536 символов. Сколько времени затратил модем на передачу файла, если известно, что для кодирования каждого символа использовалась двухбайтная кодировка. В ответе укажите целое число секунд.

3) Какие из перечисленных выражений тождественно равны выражению (A or B) and (A or C).

- A
- A or (B and C)
- B and C
- A or not (B and C)
- A and (B or C).

4) Три ученика (Лёша, Коля и Илья) умеют писать программы на трех языках программирования (C, Pascal и Python). Кто, на каких языках умеет писать программы неизвестно, но известно, что каждый ученик умеет писать программы на двух языках. При этом на Pascal умеет программировать один ученик, на Python – трое и на C – двое. Кто умеет программировать на Pascal, если известно, что Коля научился программировать на C раньше Ильи? В ответе укажите имя в именительном падеже.

5) В слове «антилопа-прыгун» уничтожается каждый второй символ, затем слово переворачивается и опять уничтожается каждый второй символ. Эти действия повторяются до тех пор, пока не останется один символ. Какой это символ?

6) Укажите через запятую в порядке возрастания все основания систем счисления, в которых число семнадцать имеет младшим разрядом двойку.

7) Дан фрагмент программы, обрабатывающей двумерный массив A размером NxN. Какой элемент массива в результате будет иметь минимальное значение, если N = 239 и первоначально все элементы массива A были равны 1000? В ответе укажите адрес искомого элемента (номер строки и номер столбца, записанные в квадратных скобках и разделенные запятой).

<i>Бейсик</i>	<i>Паскаль</i>	<i>Алгоритмический</i>
<pre> k := 0 FOR i = 1 TO N FOR j = N-i+1 TO N k = k + 1 A(i, j) = k NEXT j NEXT i </pre>	<pre> k := 0; for i := 1 to N do for j := N-i+1 to N do begin k := k + 1; A[i, j] := k; end; End; </pre>	<pre> к := 0 нц для i от 1 до N нц для j от N-i+1 до N к := к + 1 A[i, j] := к кц кц </pre>

8) Сколько записей в таблице удовлетворяют условию: «Вес >= 3 and (Г < 16 or Количество > 5)»?

Номер	Деталь	Вес	В	Ш	Г	Сложность	Количество
1	Подшипник	5	3	1	18	9	5
2	Винт	6	1	3	18	13	7
3	Болт	4	1	4	16	13	7
4	Гайка	3	6	0	15	5	2
5	Шайба	2	3	3	12	14	17
6	Ролик	3	2	4	11	13	7

9) В базе данных автосалона имеются автомобили марки Ford и марки Volkswagen следующих типов: минивэн, внедорожник, седан. Были выполнены следующие запросы к базе данных продаж автомобилей салона за текущий месяц (в период запросов данные в базе не изменялись).

<i>Запрос</i>	<i>Результат</i>
Продано всего автомобилей марки Ford	39
Продано внедорожников и седанов марки Volkswagen	36
Продано всего внедорожников и седанов	65

Какой результат выдаст следующий запрос к базе: «продано минивэнов марки Ford»?

10) У исполнителя Вычислитель три команды: умножить текущее число на 2, умножить текущее число на 3 и прибавить к текущему числу 1. За какое наименьшее количество команд можно получить число 46 из числа 1.

7. Тестовые задания для учащихся из 7–8-х классов

1) Двухзначное число в системе счисления с основанием 5 при перестановке местами цифр становится числом большим на единицу, но записанным в системе счисления с основанием 6. Найдите исходное число и представьте его в десятичной системе счисления.

2) Модему, передающему информацию со скоростью 128 байт/с, потребовалось 2 минуты и 8 секунд, для того чтобы передать растровое изображение. Укажите максимальное количество цветов в палитре изображения, если известно, что его размер составил 64x256 пикселей. Ответ записать в виде целого числа.

3) Укажите минимальное число символов в алфавите, чтобы с помощью слов из трех букв можно было бы передавать 300 различных сообщений. Слова могут содержать повторяющиеся символы. Ответ дать в виде целого числа.

4) Известно, что в методичке имеется 16 страниц, на каждой странице по 32 строки, а в каждой строке 24 символа. Вычислите информационный объем методички, если известно, что она кодировалась в двухбайтной кодировке Unicode. Ответ напишите в виде целого числа байт.

5) Есть набор точек, цвет которых задан в модели RGB:

- А = (128,5,250);
- Б = (250,250,250);
- В = (5,128,250);
- Г = (128, 128,128);
- Д = (250,128,128);
- Е = (5,5,5);
- Ж = (5, 250,5).

Выпишите последовательно точки, окрашенные в оттенки только серого цвета, расположив их в порядке возрастания яркости. В ответе укажите последовательность из выписанных подряд букв.

6) Укажите маску (шаблон), позволяющую правильно объединить в одну группу все файлы, имеющие в имени третий символ "Z", содержащие в имени после (не обязательно сразу) символ "Y", и оканчивающиеся на символ "V", имеющие расширение длиной не менее 2 символов, первый из которых "R". Под именем файла, здесь, понимается имя без расширения. Пример записи маски (шаблона) файла: ??A*.B? Знак '?' означает один любой символ. Знак '*' означает любое (возможно нулевое) количество символов.

7) База данных "Учащиеся", наряду с другими, имеет поля с названиями "пол" и "класс". В базе данных находятся записи об учащихся трех классов. Количество записей N, удовлетворяющих различным запросам, приведено в следующей таблице.

Запрос	Результат
пол=ж или класс ≠ 10	33
неверно, что (класс=9 или пол = м)	16
неверно, что (класс = 10)	24

Чему равно количество записей, удовлетворяющих запросу "класс=11 и пол = ж"?

8) Какой из тегов HTML рисует горизонтальную линию: <hr>,
, <body>, <line>?

9) Выберите наибольшее количество информации: 1 Мегабайт, 1023 Килобайта, 2000000 бит?

10) Дан массив из 6 натуральных чисел: {3,1,5,2,6,4}. Над элементами этого массива разрешено проделывать только одну операцию: менять местами любые два расположенных рядом элемента. Какое минимальное количество раз необходимо выполнить эту операцию, для того, чтобы массив оказался отсортирован по возрастанию?

ЗАДАЧИ ДЛЯ УЧАСТНИКОВ ИЗ VII–VIII КЛАССОВ

№№ 1, 2, 3, 5, 6.

ЗАДАЧИ ДЛЯ УЧАСТНИКОВ ИЗ IX–XI КЛАССОВ

№№ 2, 3, 4, 5, 7.

ТЕСТЫ К ЗАДАЧАМ И РАЗБАЛЛОВКА

Для всех задач разработано не менее 10 тестов, за каждую задачу максимальный балл составляет 100. Первые тесты используют небольшие значения в качестве входных данных, последние тесты, как правило, ориентированы на максимально допустимые значения. Поэтому неэффективные решения некоторых задач могут набирать не более 80 баллов.

Тесты и решения задач можно получить по адресу: olymp43@gmail.com.

Задачи №№6, 7 – представляют собой тестовые задания, максимальный балл за них составляет 100 баллов, то есть за каждый правильный ответ начисляется по 10 баллов. Тестовые задания не могут быть проверены автоматизированной системой, поэтому за их проверку и оценку отвечает жюри, ответственное за организацию этапа олимпиады. Ответы на тестовые задания приведены в разделе "Решения задач".

РЕШЕНИЯ ЗАДАЧ

Примеры программных решений можно получить по адресу: olymp43@gmail.com.

Далее рассматриваются ключевые идеи решения каждой задачи.

1. Лягушка

Поскольку камни не могут столкнуться и летят они с одинаковой скоростью, то всё, что требуется определить – это какая из «лягушек» приземлится ближе к краю противоположного берега. Так как длина каждого прыжка «лягушки» является постоянной, то ответ на задачу можно посчитать по формуле. Первая лягушка приземлится на расстоянии $(S-M) \bmod X$, а вторая – $(S-N) \bmod Y$.

Таким образом общее решение будет следующим:

```
S1:=(S-M) mod X;  
S2:=(S-N) mod Y;  
if S1<S2 then Write(1)  
else if S1>S2 then Write(2)  
else Write(0);
```

2. По росту становись!

Важным условием задачи является то, что ровно один ученик занял не своё место. Следовательно, мы точно знаем, что все ученики (их рост) расположены в порядке невозрастания, кроме одной позиции. Значит на одной позиции не должно выполняться условие: $a[i] \geq a[i+1]$, где a – это массив с ростом учеников. Как только такое i было найдено – это и есть ответ. Пример реализации решения на языке Паскаль:

```
i:=1;  
while (a[i]>=a[i+1]) do Inc(i);  
Write(i);
```

На самом деле массив в этой задаче не обязателен, так как мы обрабатываем только два соседних значения, то можно прямо при чтении данных сравнивать рост текущего ученика и предыдущего.

3. Частота букв

Для простого решения задачи требуется выбор подходящей структуры данных. Очевидно, что нам нужно пройти по каждому слову и запомнить, какие в нём используются буквы, при этом нас не интересует сколько раз буква входит в каждое слово, а интересует лишь сам факт вхождения. Для этих целей в Паскале удобно каждое слово представить не типом string, а множеством. Теперь, когда каждое слово представлено в виде множества букв, то достаточно найти пересечение всех таких множеств, а затем вывести все буквы результирующего множества. Стоит отметить, что для решения задачи не требуется хранить все слова в памяти, поэтому искать решение можно прямо при чтении данных. Приведём пример решения на языке Паскаль:

```

Var a, b: set of char;
    N,i,j:integer;
    c:char;
Begin
    ReadLn(N);
    a := ['a'..'z'];
    for i:=1 to N do begin
        ReadLn(s);
        b:=[];
        for j:=1 to length(s) do
            b:=b+[s[i]];
        a:=a*b;
    end;
    if a = [] then WriteLn('NO')
    else begin
        WriteLn('YES');
        for c:='a' to 'z' do
            if c in a then Write(c);
        end;
    end;
End.

```

4. Дроби

Найдём номер диагонали содержащей требуемое число. Заметим, что на i -й диагонали находится i чисел. Будем последовательно переходить от диагонали к диагонали, и уменьшать номер искомого числа. Когда этот номер будет меньше или равен количеству чисел на очередной диагонали, то необходимо остановиться.

```

read(n);
i := 0;
while n > i + 1 do
begin
    n := n - (i + 1);
    i := i + 1;
end;
i := i + 1;

```

После этого осталось найти n -е число на i -й диагонали. На этой диагонали находятся числа от $1/i$ до $i/1$. Номер определяется в зависимости от чётности диагонали с помощью следующего условия:

```

if i mod 2 = 0 then
    writeln(i - n + 1, '/', n)
else
    writeln(n, '/', i - n + 1);

```

Это решение набирает 80 баллов. Чтобы набрать 100 баллов заметим, что можно сразу рассмотреть большинство диагоналей, воспользовавшись формулой $1+2+3+...+i=i*(i+1)/2$. Мы знаем, что если число стоит на i -й диагонали, то $i*(i-1)/2 \leq n < i*(i+1)/2$. Поэтому можно начать перебор диагоналей с $\sqrt{2n}-1$. Полное решение:

```

var n, i : int64;
begin
    read(n);
    i := round(sqrt(2.0 * n)) - 1;
    n := n - (i * (i + 1) div 2);
    while n > i + 1 do
    begin
        n := n - (i + 1);
        i := i + 1;
    end;
    i := i + 1;
    if i mod 2 = 0 then
        writeln(i - n + 1, '/', n)
    else
        writeln(n, '/', i - n + 1);
    end.

```

5. Кратные числа

Заведём массив *u* типа *boolean*. Здесь *u[i]* соответствует тому обведено число или нет. Изначально все *u[i]* равны *false*. Будем последовательно выполнять описанные шаги. Чтобы эффективно обвести все числа кратные некоторому числу *x* можно воспользоваться следующим циклом:

```
j := x;
while j <= n do
begin
  u[j] := true;
  j := j + x;
end;
```

Это решение работает, но не набирает полный балл. Для того, чтобы набрать полный балл необходимо заметить, что можно поменять порядок шагов и идти от меньших чисел к большим. Например, пусть необходимо обвести числа кратные 8, 2 и 3. Обведём сначала числа кратные 2, потом 3. Теперь необходимо обвести числа, кратные 8, но поскольку само число 8 уже обведено, то этот шаг можно пропустить – количество обведённых чисел не изменится.

Возможный вариант реализации.

```
var u, a : array[1..300000] of boolean;
    n, k, i, j : integer;
begin
  read(n, k);
  for i := 1 to k do
  begin
    read(j);
    a[j] := true;
  end;

  for i := 1 to 300000 do
    if (a[i] = true) and (u[i] = false) then
    begin
      j := i;
      while j <= n do
      begin
        u[j] := true;
        j := j + i;
      end;
    end;

    j := 0;
    for i := 1 to n do
      if u[i] = false then
        j := j + 1;
    writeln(j);
  end.
```

Для упорядочивания чисел используется вспомогательный массив *a*, где *a[i] = true*, если число *i* необходимо вычёркивать.

6.

1	2	3	4	5	6	7	8	9	10
2050 байт, 2100 байт	8	б)	Лёша	р	3, 5, 15	[1, 239]	4	10	6

7.

1	2	3	4	5	6	7	8	9	10
6	256	7	24576	ЕГБ	??Z*Y*V.R?* или ??Z*Y*V.R*?	7	<hr>	1 Мега- байт	5