

Муниципальный этап
Всероссийской олимпиады школьников
по информатике

в 2014 – 2015 учебном году

Разборы решений и идеи тестов

Муниципальный этап Всероссийской олимпиады
школьников по информатике
в 2014 – 2015 учебном году
8 класс

Время выполнения задач — 4 часа

Ограничение по времени — 2 секунды на тест

Ограничение по памяти — 64 мегабайта

8.1. «Носим кирпичи». Семья Торопыжкиных строит на даче новый сарай, для чего привезено n кирпичей. Петя Торопыжкин за один раз переносит не более m кирпичей, а его брат — не более k кирпичей. Переноской кирпичей будет заниматься кто-то один из них. Какое минимальное количество ходок нужно будет сделать для переноски всех имеющихся кирпичей?

Формат входа: В единственной строке через пробел заданы три целых числа n , m и k ($1 \leq n, m, k \leq 10^4$).

Формат выхода: Выдайте единственное целое число – минимальное количество ходок, за которое один из братьев перенесёт все кирпичи.

Пример

Вход: Выход:

13 7 4 2

8.2. «Решаем задачи». В течение октября, готовясь к районному туру олимпиады по информатике, Петя Торопыжкин решает задачи. В n -й день октября ($1 \leq n \leq 31$) он прорешивает $k(n) = an^2 + bn + c$ задач. Если коэффициенты таковы, что в какой-то день значение $k(n)$ отрицательно, то это означает, что в этот день он не решил ни одной задачи. По заданным целым коэффициентам a , b , c найдите, сколько задач прорешал Петя за тридцать один день октября.

Формат входа: В единственной строке через пробел заданы три целых числа a , b , c ; каждое по модулю не превосходит 1000.

Формат выхода: Выведите единственное целое неотрицательное число — количество задач, решённых Петей.

Пример

Вход: Выход:

1 -30 200 1901

8.3. «Книги в макулатуру». В результате сбора макулатуры, проведённого в школе у Пети Торопыжкина, кроме всей прочей бумажной продукции было собрано n книг. Таская их в машину, Петя обнаружил, что их нечётное количество, и задумался о том, сколько всего страниц в сумме в самой тонкой книге, в самой толстой и в самой «средней» (то есть в книге, оказавшейся бы в середине ряда,

стоит покупать торт, и длину маршрута, который придётся пройти Пете. Номера магазинов должны соответствовать исходному списку; нумерация начинается от 1. Если возможных ответов несколько, выдайте любой из них.

Пример

<u>Вход:</u>	<u>Выход:</u>
1 2	3 4 24
4	
3 7 30 70	
-1 -10 60 60	
5 0 30 100	
0 7 40 60	

8.5. «Степень строки». n -й *степенью строки* в математике называют строку, получаемую n -кратным записыванием данной строки. Например, строка ABCD во второй степени — это ABCDABCD, а XY в четвёртой — это XYXYXYXY. Заметим, что строку XYXYXYXY также можно рассматривать как квадрат строки XYXY. Для данной строки определите, является ли она какой-либо степенью какой-либо более короткой строки.

Формат входа: В единственной строке задана исходная непустая строка, имеющая длину не более 255 символов и состоящая только из заглавных символов латиницы.

Формат выхода: Если данная строка не является никакой степенью никакой более короткой строки, то нужно вывести -1. Если она является степенью, то нужно вывести эту степень и через пробел — строку-основание. Если есть несколько ответов, то нужно вывести тот, у которого степень наибольшая.

Пример 1

<u>Вход:</u>	<u>Выход:</u>
ABCABD	-1

Пример 2

<u>Вход:</u>	<u>Выход:</u>
XYXYXYXY	4 XY

**Муниципальный этап Всероссийской олимпиады
школьников по информатике
в 2014 – 2015 учебном году
8 класс. Разбор решений и идеи тестов**

Далее в разборах используются обозначения типов: short int — 2-байтовый целый тип (integer в TurboPascal/BorlandPascal и int в BorlandC++), int — 4-байтовый целый тип (longint в TurboPascal/BorlandPascal/FreePascal, long int в BorlandC++, integer в Delphi/PascalABC и int в VisualC++/C++ Builder), int64 — 8-байтный целый (отсутствует в BorlandPascal и BorlandC++, но присутствует в системах программирования под Windows).

Задачи оцениваются по 100 баллов за каждую; стало быть, полная стоимость пакета — 500 баллов. В рамках одной задачи все тесты считать равноценными.

8.1. «Носим кирпичи». *Семья Торопыжских строит на даче новый сарай, для чего привезено n кирпичей. Петя Торопыжский за один раз переносит не более m кирпичей, а его брат — не более k кирпичей. Переноской кирпичей будет заниматься кто-то один из них. Какое минимальное количество ходок нужно будет сделать для переноски всех имеющихся кирпичей?*

Программный комитет оценивает эту задачу как весьма простую. Вычисляем количество ходок, которые надо сделать Пете, количество ходок, которые нужно сделать брату, и выбираем меньшее из этих чисел. Некоторая тонкость состоит в вычислении количества ходок; если выражать математически, то это будут величины $\lceil n/m \rceil$ и $\lceil n/k \rceil$, где $\lceil \cdot \rceil$ — операция округления вверх. В большинстве языков эта операция реализована, однако лучше не прибегать к вещественным вычислениям и написать код примерно такой:

```
res := n div m;  
if n mod m <> 0 then res := res+1;
```

Идеи тестов:

1. Минимальный тест $n = m = k = 1$;
2. Минимальный тест $n = 10, m = k = 1$;
3. Минимальный тест $n = m = 1, k = 10$;
4. Минимальный тест $n = 1, m = 10, k = 1$;
5. Максимальный тест $n = m = k = 10^4$;
6. Максимальный тест $n = 10, m = k = 10^4$;
7. Максимальный тест $n = m = 10^4, k = 10$;
8. Максимальный тест $n = 10^4, m = 10, k = 10^4$;
- 9–20. Случайные тесты.

8.2. «Решаем задачи». *В течение октября, готовясь к районному туру олимпиады по информатике, Петя Торопыжский решает задачи. В n -й день октября*

($1 \leq n \leq 31$) он прорешивает $k(n) = an^2 + bn + c$ задач. Если коэффициенты таковы, что в какой-то день значение $k(n)$ отрицательно, то это означает, что в этот день он не решил ни одной задачи. По заданным целым коэффициентам a, b, c найдите, сколько задач прорешал Петя за тридцать один день октября.

Данная задача представляется простой: для её решения необходимо аккуратно организовать цикл суммирования величин, вычисляемых по указанной формуле, с проверкой их неотрицательности.

Идеи тестов:

1. Зависимость — константа, для всех n значения отрицательны: $a = b = 0$, $c = -1$;
2. Зависимость возрастающая линейная, для всех n значения отрицательны: $a = 0$, $b = 1$, $c = -100$;
3. Зависимость убывающая линейная, для всех n значения отрицательны: $a = 0$, $b = -1$, $c = -100$;
4. Зависимость квадратичная, «рога вверх», для всех n значения отрицательны: $a = 1$, $b = -33$, $c = -70$;
5. Зависимость квадратичная, «рога вниз», для всех n значения отрицательны: $a = -1$, $b = -33$, $c = -70$;
6. Зависимость — константа, для всех n значения нулевые: $a = b = c = 0$;
7. Зависимость возрастающая линейная, для некоторых n значения отрицательны: $a = 0$, $b = 1$, $c = -10$;
8. Зависимость убывающая линейная, для некоторых n значения отрицательны: $a = 0$, $b = -1$, $c = 10$;
9. Зависимость возрастающая линейная, для всех n значения положительны: $a = 0$, $b = 1$, $c = 10$;
10. Зависимость убывающая линейная, для всех n значения положительны: $a = 0$, $b = -1$, $c = 100$;
11. Зависимость квадратичная, «рога вверх», промежуток отрицательных значений в начале: $a = 2$, $b = -24$, $c = -90$;
12. Зависимость квадратичная, «рога вверх», промежуток отрицательных значений в середине: $a = 2$, $b = -66$, $c = 400$;
13. Зависимость квадратичная, «рога вверх», промежуток отрицательных значений в конце: $a = 3$, $b = -121$, $c = 560$;
14. Зависимость квадратичная, «рога вверх», все значения положительны: $a = 1$, $b = 3$, $c = 1$;
15. Зависимость квадратичная, «рога вниз», промежуток отрицательных значений в начале: $a = -3$, $b = 121$, $c = -560$;
16. Зависимость квадратичная, «рога вниз», промежуток отрицательных значений в середине: $a = -2$, $b = 66$, $c = -400$;
17. Зависимость квадратичная, «рога вниз», промежуток отрицательных значений в конце: $a = -2$, $b = 24$, $c = 90$;

18. Зависимость квадратичная, «рога вниз», все значения положительны:
 $a = -1, b = 33, c = 70$;
19. Максимальный тест, $a = b = c = -1000$;
20. Максимальный тест, $a = b = c = 1000$;
- 21–25. Случайные большие тесты.

Примечание: Ответы к тестам с 1 по 18 входят в `short int`. Прочие тесты подразумевают ответ, входящий в `int`.

8.3. «Книги в макулатуру». *В результате сбора макулатуры, проведённого в школе у Пети Торопыжкина, кроме всей прочей бумажной продукции было собрано n книг. Таская их в машину, Петя обнаружил, что их нечётное количество, и задумался о том, сколько всего страниц в сумме в самой тонкой книге, в самой толстой и в самой «средней» (то есть в книге, оказавшейся бы в середине ряда, упорядоченного по толщине). Конечно, таская книги, он узнал, сколько страниц в каждой из них.*

Идейно данная задача является лёгкой идейно, однако для её решения требуется привлечь процедуру сортировки, что делает её для восьмиклассников средней по сложности.

Идея решения крайне проста: отсортируем (по возрастанию или по убыванию) полученный массив чисел и найдём сумму первого, последнего и $(n + 1)/2$ -го элементов (при нумерации элементов массива от единицы; при нумерации от нуля надо брать элемент с индексом $(n - 1)/2$).

Следует также отметить, что при размерах сортируемого массива более 14000–15000 не будут проходить по времени «наивные» алгоритмы сортировки, имеющие квадратичную сложность: сортировка пузырьком, сортировка вставкой и т.п. Полный балл можно получить, используя «умные» алгоритмы, например, быструю сортировку (сортировку Хоара), или же используя библиотечные функции сортировки (в которых, собственно, как раз реализована быстрая сортировка). Те, кто будут использовать неоптимальные алгоритмы, получают около 65% полного балла.

Альтернативный путь решения, более оптимальный по времени работы и более простой по реализации, хотя и идейно более сложный, состоит в использовании идеи линейной сортировки. Поскольку количество различных объектов (возможных количеств страниц в книге) не так велико, то создав массив $a[\cdot]$ из 1500 элементов, можно заполнить его ячейки за время, пропорциональное количеству книг (то есть n). В i -й ячейке этого массива хранится число книг с количеством страниц i . Поэтому заполнение этого массива состоит в том, чтобы после прочтения количества страниц в очередной книжке увеличить на 1 соответствующую ячейку массива (вначале массив $a[\cdot]$ заполнен нулями).

После заполнения массива имеем, что количество страниц в самой тонкой книге — это номер первой ненулевой ячейки массива, а количество страниц в самой

толстой — номер последней ненулевой ячейки. Некоторое затруднение вызывает нахождение количества страниц у «средней» книги, поскольку в отличие от предыдущего пути решения у нас нет явной упорядоченной последовательности книг.

Для нахождения этой величины надо найти ячейку с номером i^* таким, что

$$\sum_{k=1}^{i^*-1} a[k] < (n+1)/2 \quad \text{и} \quad \sum_{k=1}^{i^*} a[k] \geq (n+1)/2,$$

то есть ячейку, содержащую «среднюю» книгу. Этот поиск можно провести, суммируя последовательно элементы массива $a[\cdot]$ и остановив перебор элементов, когда выполнится второе неравенство.

Идеи тестов:

1. Минимальный тест, $n = 3$, все элементы одинаковы;
2. Минимальный тест, $n = 3$, все элементы различны;
3. $n = 11$, все элементы одинаковы;
4. $n = 11$, все элементы различны;
5. $n = 11$, есть одинаковые элементы;
- 6–13. Случайные тесты, позволяющие использовать «наивные» алгоритмы сортировки;
14. Максимальный тест, $n = 99999$, все элементы одинаковы;
15. Максимальный тест, $n = 99999$, не все элементы одинаковы;
- 16–20. Случайные тесты, требующие использования оптимальных сортировок.

8.4. «Прогулка по магазинам». Родители поручили Пете Торопыжскину купить к празднику колбасу и торт, выдав на руки некоторую достаточно большую сумму денег и сказав, что сдачу Петя может забрать себе. Поэтому интерес Пети состоит в том, чтобы купить нужные продукты подешевле (конечно же, не в ущерб качеству!). Около его дома расположено n магазинов, цены на колбасу и торты в которых Пете известны. Ему нужно узнать длину кольцевого маршрута, по которому ему придётся пройти от дома до магазина (или двух магазинов, если дешёвая колбаса продаётся не там же, где дешёвые торты) и обратно до дома. Петя не ограничен во времени, так что ему нет нужды находить самый короткий маршрут.

Следует отметить, что поскольку Петя живёт в городе, то он вынужден перемещаться между двумя точками не напрямик, а по прямоугольной сетке улиц. Поэтому для вычисления расстояния между двумя точками $A(x_a, y_a)$ и $B(x_b, y_b)$ применяется формула так называемого манхэттенского расстояния: $d_1(A, B) = |x_a - x_b| + |y_a - y_b|$. Действительно: сначала надо по улице, параллельной оси абсцисс, пройти расстояние $|x_a - x_b|$, а потом по улице, параллельной оси ординат, — расстояние $|y_a - y_b|$. Сочетание «манхэттенское расстояние»

происходит от того, что такой сеткой улиц обладает Манхэттен, район города Нью-Йорка.

Программный комитет считает, что эта задача имеет сложность несколько выше средней.

Центральной идеей здесь является применение алгоритма поиска минимума в массиве. Считываем данные, которые можно хранить или в четырёх параллельных массивах, или в массиве структур. Затем за один проход ищем индекс i элемента, хранящего минимальное значение цены на колбасу, и индекс j элемента, хранящего минимальную цену на торты. Затем с использованием указанной формулы ищем значение

$$d = (|x_0 - x_i| + |y_0 - y_i|) + (|x_0 - x_j| + |y_0 - y_j|) + (|x_j - x_i| + |y_j - y_i|)$$

— длину в пути в требуемой метрике. Затем выводим величины i, j, d (корректируя значения индексов, если в используемом языке нумерация элементов идёт от нуля).

Идеи тестов:

1. Минимальный тест — один магазин, $n = 1$;
2. $n = 2$, минимальные цены на колбасу и торты в одном магазине;
3. $n = 2$, минимальные цены на колбасу и торты в разных магазинах;
4. $n = 10$, все цены разные, минимальные цены в одном магазине;
5. $n = 10$, все цены разные, минимальные цены в разных магазинах;
6. $n = 10$, есть несколько магазинов с минимальными ценами;
- 7–12. то же, что в тестах 1–6, но расстояние не входит в short int;
- 13–17. случайные тесты, расстояние входит в short int;
- 18–20. случайные тесты, расстояние не входит в short int.

8.5. «Степень строки». n -й степенью строки в математике называют строку, получаемую n -кратным записыванием данной строки. Например, строка $ABCD$ во второй степени — это $ABCDABCD$, а XY в четвёртой — это $XYXYXYXY$. Заметим, что строку $XYXYXYXY$ также можно рассматривать как квадрат строки $XYXY$. Для данной строки определите, является ли она какой-либо степенью какой-либо более короткой строки.

По мнению программного комитета, данная задача является сложной для восьмиклассников, так как требует работы со строками и реализации нестандартного алгоритма.

В целом, подразумевается «лобовое» решение, в принципе, доступное для реализации восьмиклассниками. Идея состоит в следующем: перебираем возможные длины d базовой строки от 1 до $\lfloor l/2 \rfloor$, где l — длина проверяемой строки **str**, а $\lfloor \cdot \rfloor$ — округление вниз. Для всякой тестируемой длины d нужно проверить, что

длина l тестируемой строки кратна d , и кроме того, что для всех индексов i от 1 до d выполняется совпадение символов

$$\text{str}[i] = \text{str}[i + d] = \text{str}[i + 2d] = \dots = \text{str}[i + kd],$$

где $k = l/d - 1$. (В случае отсчёта символов строки от нуля следует скорректировать эту формулу.) Если это равенство верно для всех i , то проверяемая строка является $(k+1)$ -й степенью строки, состоящей из первых d символов. Если ни для какого d из указанного диапазона такое совпадение не верно хотя бы для одного i , то проверяемая строка степенью не является, о чём и следует сообщить.

При заданных ограничениях на входные данные «любовой» алгоритм укладывается в ограничения по времени.

Идеи тестов:

1. минимальный тест — односимвольная строка;
2. минимальный тест — двухсимвольная строка из одинаковых символов;
3. минимальный тест — двухсимвольная строка из разных символов;
4. шестисимвольная строка, являющаяся квадратом;
5. шестисимвольная строка, являющаяся кубом;
6. шестисимвольная строка, отличающаяся от куба в 3-м символе;
7. шестисимвольная строка, отличающаяся от куба в 4-м символе;
8. 64-символьная строка, являющаяся 8-й степенью;
9. 64-символьная строка, являющаяся квадратом;
10. максимальный тест, 255-символьная строка, не являющаяся степенью;
11. максимальный тест, 255-символьная строка, являющаяся 3-й степенью;
12. максимальный тест, 255-символьная строка, являющаяся 5-й степенью;
13. максимальный тест, 255-символьная строка, являющаяся 15-й степенью;
14. максимальный тест, 255-символьная строка, являющаяся 17-й степенью;
15. максимальный тест, 255-символьная строка, являющаяся 51-й степенью;
16. максимальный тест, 255-символьная строка, являющаяся 85-й степенью;
17. максимальный тест, 255-символьная строка, являющаяся 255-й степенью;
- 18–22. случайные тесты, являющиеся степенью;
- 23–25. случайные тесты, не являющиеся степенью.