

3-4 курс

Олимпиада «Интеллект». Секция «Программирование».

Внимание! Для всех задач действуют следующие ограничения: ограничение времени — 1 секунда, ограничение памяти — 16 мб.

Задача 1. Практичные числа

Практичное число — это натуральное число n , такое что все меньшие натуральные числа могут быть представлены в виде суммы различных делителей числа n . Например, 12 является практичным числом, поскольку все числа от 1 до 11 можно представить в виде суммы делителей 1, 2, 3, 4 и 6 этого числа (сами делители равны самим себе):

$$5 = 3 + 2; 7 = 6 + 1; 8 = 6 + 2; 9 = 6 + 3; 10 = 6 + 3 + 1; 11 = 6 + 3 + 2.$$

Необходимо написать программу, которая считает количество практичных чисел, меньших N .

Формат входных данных

В первой и единственной строке файла “INPUT.TXT” находится число N ($1 \leq N \leq 200$).

Формат выходных данных

Требуется вывести в файл “OUTPUT.TXT” количество практичных чисел, меньших N .

Пример:

Формат входных данных
15
Формат выходных данных
6

Задача 2. Рекламный баннер

Лагерь «Интеллект» решил разместить на фасаде здания рекламные баннеры. Фасад представляет собой прямоугольник размера $M \times N$ квадратных плиток. Рекламный баннер представляет собой прямоугольник $H \times W$ размера плитки и размещается на фасаде ровно по границам плиток.

Был проведён аукцион, в котором всем желающим предлагалось указать положение баннера на фасаде в виде координат левого нижнего угла и размера баннера в единицах длины плитки и цену, которую компания-рекламодатель готова заплатить, если баннер будет виден полностью. Если баннер будет виден не полностью, его цена уменьшается пропорционально видимой площади.

Необходимо разместить баннеры на фасаде так, чтобы максимизировать рекламную выручку. Баннеры могут перекрывать друг друга, то есть если баннер компании «Тодо» закроет собой часть баннера компании «Додо», то компания «Додо» заплатит за рекламу меньше пропорционально закрытой площади. Передвигать баннеры на другие позиции нельзя.

Формат входных данных

В первой строке файла “INPUT.TXT” записаны два целых числа M и N ($1 \leq M; N \leq 1000$ - размер фасада, выраженный в единицах длины плитки. M - размер по вертикали (число рядов), N - размер по горизонтали (число столбцов) и целое число K ($0 \leq K \leq 6000$) - количество компаний, подавших заявки на участие в аукционе. В последующих K строках для каждой компании подаются пять чисел: $R; C; H; W; P$, где R - номер ряда левого нижнего угла баннера, C - номер столбца левого нижнего угла баннера. Левый нижний угол фасада здания имеет координаты $(0; 0)$. H - высота баннера в единицах длины плитки, W - ширина баннера в единицах длины плитки. $R; C; H; W$ - целые неотрицательные числа, допустимые для заданного размера фасада. P - целое число ($1 \leq P$

3-4 курс

Олимпиада «Интеллект». Секция «Программирование».

Внимание! Для всех задач действуют следующие ограничения: ограничение времени — 1 секунда, ограничение памяти — 16 мб.

$\leq 2^{31} - 1$), указывающее цену, которую компания готова заплатить за баннер.

Формат выходных данных

Требуется вывести в файл “OUTPUT.TXT” максимальную возможную рекламную выручку.

Пример:

Формат входных данных
1000 1000 2
0 0 500 1000 1000001
0 0 1000 500 1000000
Формат выходных данных
1500001

Формат входных данных
10 10 1
0 0 10 10 1
Формат выходных данных
1

Формат входных данных
10 10 10
0 0 1 1 1
1 1 1 1 2
2 2 1 1 3
3 3 1 1 4
4 4 1 1 5
5 5 1 1 6

6 6 1 1 7
7 7 1 1 8
8 8 1 1 9
9 1 1 1 10
Формат выходных данных
55

Задача 3. Повреждённый XML

Формат XML является распространенным способом обмена данными между различными программами. Недавно Егор написал небольшую программу, которая сохраняет некоторую важную информацию в виде XML-строки.

XML-строка состоит из открывающих и закрывающих тегов. Открывающий тег начинается с открывающей угловой скобки (<), за ней следует имя тега — непустая строка из строчных букв английского алфавита, а затем закрывающая угловая скобка (>). Примеры открывающих тегов: <a>, <dog>.

Закрывающий тег начинается с открывающей угловой скобки, за ней следует прямой слеш (/), затем имя тега — непустая строка из строчных букв английского алфавита, а затем закрывающая угловая скобка. Примеры закрывающихся тегов: , </dog>.

XML-строка называется корректной, если она может быть получена по следующим правилам:

- Пустая строка является корректной XML-строкой.
- Если A и B — корректные XML-строки, то строка AB, получающаяся приписыванием строки B в конец строки A, также является корректной XML-строкой.

3-4 курс

Олимпиада «Интеллект». Секция «Программирование».

Внимание! Для всех задач действуют следующие ограничения: ограничение времени — 1 секунда, ограничение памяти — 16 мб.

- Если A — корректная XML-строка, то строка $\langle X \rangle A \langle /X \rangle$, получающаяся приписыванием в начало A открывающегося тега, а в конец — закрывающегося с таким же именем, также является корректной XML-строкой. Здесь X — любая непустая строка из строчных букв английского алфавита.

Например, представленные ниже строки:

```
<a></a>
<a><ab></ab><c></c></a>
```

являются корректными XML-строками, а такие строки как:

```
<a></b>
<a><b></a></b>
```

не являются корректными XML-строками.

Егор отправил файл с сохраненной XML-строкой по электронной почте своему коллеге Вадиму. К сожалению, файл повредился: ровно один символ в строке заменился на некоторый другой символ.

Требуется написать программу, которая по строке, которую получил Вадим, восстановит исходную XML-строку, которую отправлял Егор.

Формат входных данных

Входной файл “INPUT.TXT” содержит одну строку. Длина строки лежит в пределах от 7 до 1000, включительно. Строка содержит только строчные буквы английского алфавита и символы «<» (ASCII код 60), «>» (ASCII код 62) и «/» (ASCII код 47). Строка заканчивается переводом строки.

Формат выходных данных

Выходной файл “OUTPUT.TXT” должен содержать корректную XML-строку, которая может быть получена из строки во входном файле заменой ровно одного символа на другой. Если вариантов ответа несколько, можно вывести любой.

Формат входных данных
<a>
Формат выходных данных
<a>

Формат входных данных
<a/>
Формат выходных данных
<a>

Задача 4. Индийская клавиатура

Тимур недавно устроился стажером в индийскую IT-компанию. Его обязанности как стажера — оцифровывать тексты с изречениями великих индийских мудрецов. Добравшись до своего рабочего места, Тимур был весьма неприятно удивлен — оказалось, что вместо привычных нам клавиатур индийские программисты используют огромную прямоугольную клавиатуру размера $n \times m$, в каждой ячейке которой содержится уникальная буква из некоторого цифрового алфавита, имеющая номер от 1 до $n * m$.

Окинуть всю клавиатуру взглядом может быть затруднительно, поэтому в любой момент времени Тимур доступен лишь некоторый прямоугольник размера $p \times q$ (который он волен выбрать по своему желанию). Разумеется, при печати Тимур может использовать только те буквы, которые

3-4 курс

Олимпиада «Интеллект». Секция «Программирование».

Внимание! Для всех задач действуют следующие ограничения: ограничение времени — 1 секунда, ограничение памяти — 16 мб.

попадают в выбранный им прямоугольник. Более того, за одно действие он может переконфигурировать клавиатуру: изменить прямоугольник, тем самым изменив набор доступных букв.

Изначальную конфигурацию Тимур выбирает на свое усмотрение (но на это не тратится дополнительное действие).

Изречение, которое ему предстоит напечатать сегодня, состоит из s букв, принадлежащих тому же цифровому алфавиту, что и буквы на клавиатуре. В целях оптимизации процесса Тимуру хочется как можно меньше раз перенастраивать клавиатуру в процессе набора текста. Помогите ему посчитать, сколько раз ему придется это делать, если действовать оптимально.

Формат входных данных

В первой строке файла “INPUT.TXT” содержится строка, содержащая четыре числа n, m, p, q ($1 \leq p \leq n \leq 500$; $1 \leq q \leq m \leq 500$) — размеры клавиатуры и размеры участка клавиатуры, который видит Тимур.

В следующих n строках задано по m чисел — буквы в соответствующих ячейках клавиатуры.

Далее следует число s ($1 \leq s \leq 3 * 10^5$) — размер текста, который предстоит набрать Тимур.

В последней строке через пробел следуют s букв a_i — сам текст.

Гарантируется, что $1 \leq a_i \leq n * m$.

Формат выходных данных

В файл “OUTPUT.TXT” выведите одно число — минимальное количество переконфигураций клавиатуры.

5
1 1 3 2 3
Формат выходных данных
3

В этом примере в качестве начальной конфигурации Тимур может взять прямоугольник размера 1×1 , содержащий букву 1.

Тогда первые две буквы он сможет напечатать без перенастройки клавиатуры, а затем ему придется трижды последовательно перенастраивать клавиатуру на прямоугольники с буквами 3, 2 и снова 3.

Формат входных данных

2 2 1 1
1 3
2 4