

Олимпиада "Ломоносов" по информатике. 2016-17.
5-9 класс. Отборочный тур.

Задача 1.1: Последовательность

Последовательность целых чисел $\{a_n\}$ задается следующим рекуррентным соотношением:

$$a_n = 2 * a_{n-3} - a_{n-2} + a_{n-1},$$

$$a_1 = 1,$$

$$a_2 = 1,$$

$$a_3 = 1.$$

Все числа последовательности выписали в строку. Отправьте на проверку число с 114 по 187 символ (символы нумеруются с единицы).

Задача 1.2: Последовательность

Последовательность целых чисел $\{a_n\}$ задается следующим рекуррентным соотношением:

$$a_n = 2 * a_{n-3} - a_{n-2} + a_{n-1},$$

$$a_1 = 0,$$

$$a_2 = 1,$$

$$a_3 = 1.$$

Все числа последовательности выписали в строку. Отправьте на проверку число с 124 по 178 символ (символы нумеруются с единицы).

Задача 1.3: Последовательность

Последовательность целых чисел $\{a_n\}$ задается следующим рекуррентным соотношением:

$$a_n = 2 * a_{n-3} - a_{n-2} + a_{n-1},$$

$$a_1 = 1,$$

$$a_2 = 0,$$

$$a_3 = 1.$$

Все числа последовательности выписали в строку. Отправьте на проверку число с 102 по 165 символ (символы нумеруются с единицы).

Задача 1.4: Последовательность

Последовательность целых чисел $\{a_n\}$ задается следующим рекуррентным соотношением:

$$a_n = 2 * a_{n-3} - a_{n-2} + a_{n-1},$$

$a_1 = 1,$
 $a_2 = 1,$
 $a_3 = 0.$

Все числа последовательности выписали в строку. Отправьте на проверку число с 110 по 196 символ (символы нумеруются с единицы).

Задача 2: Запись числа

На вход программе подаётся положительное целое число N ($N < 1000001$) и последовательность из N символов. Требуется определить, можно ли из введённой последовательности выбрать символы (в любом порядке) так, чтобы получилась запись шестнадцатеричного числа по правилам языка Си. В языке Си запись шестнадцатеричного числа начинается с нуля (0), за ним следует строчная буква икс (x), за которой следуют одна или более чем одна шестнадцатеричная цифра (любая из: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). Составлять запись числа можно только из тех “буквенных” цифр, которые подходят по регистру, то есть, являются заглавными. Любая цифра в записи числа может быть использована не большее количество раз, чем количество её вхождений во введённую последовательность. Если можно составить запись хотя бы для одного числа, то нужно вывести запись для наибольшего из возможных чисел. Если запись составить нельзя, программа выводит No

Примеры

Входные данные

17
I-AM-A-PROGRAMMER

Выходные данные

No

Входные данные

17
1xAMxAxPR0GRaMMER

Выходные данные

0xEAA1

Задача 3.1: Художник

Для создания новой картины мальчик Казимир взял клетчатый лист бумаги размера 3×6 . Затем он выбрал произвольную клетку и покрасил её в черный цвет. Каждую следующую клетку он красил так, чтобы у нее была ровно одна покрашенная соседняя клетка (по сторонам). Так продолжалось до тех пор, пока были клетки котые можно покрасить по этому правилу.

Вы решили повторить работу Казимира и вам требуется получить все варианты картин, которые могли получиться.

Например на поле 2×2 могло получиться 4 картины

01
11

10

11

11
10

11
01

Ответ отправьте в формате как в примере. Картины разделяются пустой строкой. Покрашенная клетка выводится символом 1, пустая 0.

Задача 3.2: Художник

Для создания новой картины мальчик Казимир взял клетчатый лист бумаги размера 4×5 . Затем он выбрал произвольную клетку и покрасил её в черный цвет. Каждую следующую клетку он красил так, чтобы у нее была ровно одна покрашенная соседняя клетка (по сторонам). Так продолжалось до тех пор, пока были клетки, которые можно покрасить по этому правилу.

Вы решили повторить работу Казимира и вам требуется получить все варианты картин, которые могли получиться.

Например на поле 2×2 могло получиться 4 картины

01
11

10
11

11
10

11
01

Ответ отправьте в формате как в примере. Картины разделяются пустой строкой. Покрашенная клетка выводится символом 1, пустая 0.

Задача 3.3: Художник

Для создания новой картины мальчик Казимир взял клетчатый лист бумаги размера 6×3 . Затем он выбрал произвольную клетку и покрасил её в черный цвет. Каждую следующую клетку он красил так, чтобы у нее была ровно одна покрашенная соседняя клетка (по сторонам). Так продолжалось до тех пор, пока были клетки, которые можно покрасить по этому правилу.

Вы решили повторить работу Казимира и вам требуется получить все варианты картин, которые могли получиться.

Например на поле 2×2 могло получиться 4 картины

01
11

10
11

11
10

11
01

Ответ отправьте в формате как в примере. Картины разделяются пустой строкой. Покрашенная клетка выводится символом 1, пустая 0.

Задача 3.4: Художник

Для создания новой картины мальчик Казимир взял клетчатый лист бумаги размера 5×4 . Затем он выбрал произвольную клетку и покрасил её в черный цвет. Каждую следующую клетку он красил так, чтобы у нее была ровно одна покрашенная соседняя клетка (по сторонам). Так продолжалось до тех пор, пока были клетки, которые можно покрасить по этому правилу.

Вы решили повторить работу Казимира и вам требуется получить все варианты картин, которые могли получиться.

Например на поле 2×2 могло получиться 4 картины

01
11

10
11

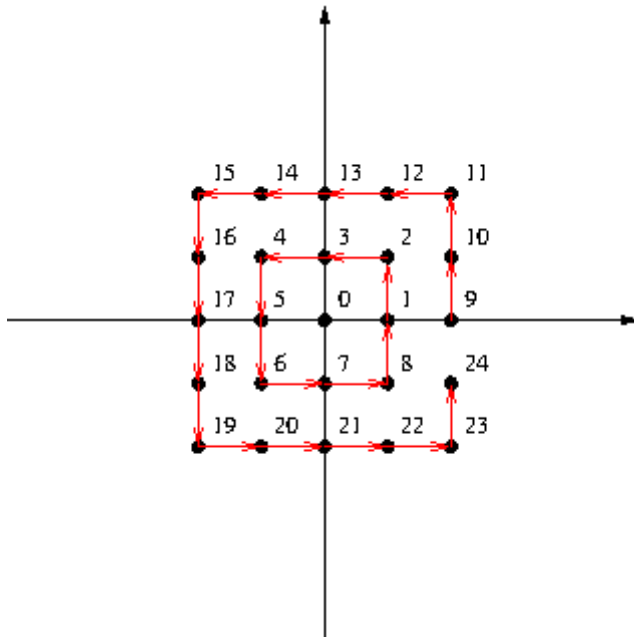
11
10

11
01

Ответ отправьте в формате как в примере. Картины разделяются пустой строкой. Покрашенная клетка выводится символом 1, пустая 0.

Задача 4: Прямые и точки

Рассмотрим координатную плоскость. Занумеруем точки с целочисленными координатами следующим образом. Точка $(0,0)$ получает номер 0. Точка $(1,0)$ получает номер 1, $(1,1)$ получает номер 2, $(0,1)$ получает номер 3 и так далее против часовой стрелки.



Напишите программу, определяющую принадлежность точек прямой. Сначала на стандартном потоке ввода задаются два номера точек, определяющие прямую на плоскости. Точки не совпадают. Затем задается последовательность номеров точек, принадлежность которых заданной прямой требуется проверить. Последовательность заканчивается числом -1. Для каждой точки в последовательности на стандартный поток вывода напечатайте 0, если эта точка не находится на заданной прямой, и 1, если точка находится на прямой. Все номера точек - 32-битные положительные целые числа. Число проверяемых точек во входном потоке не превышает 500000.

Примеры

Входные данные

```
16 2 12 3 6 -1
```

Выходные данные

```
0 1 0
```

Задача 5: Сжатие изображения

Важный раздел информатики - это алгоритмы сжатия изображений. В этой задаче мы будем рассматривать только сжатие черно-белых картинок без потерь. Задача помехоустойчивого кодирования заключается в том, что входные бинарные данные преобразовываются в закодированные бинарные данные таким образом, чтобы в размер сжатых данных был минимальным. Разработайте алгоритм и напишите программу, которая реализовывает наиболее эффективное по вашему мнению алгоритм сжатия и распаковки черно-белых изображений.

На вход программе подается либо изображение для сжатия, либо поток данных для распаковки, а именно: на стандартном потоке ввода (т. е., например, с клавиатуры) программе сначала задается число 0, если программа должна сжать данные, либо число 1, если программа должна распаковать данные. В случае кодирования далее на вход подается картинка 51 строк по 100 символов. Цифра '0' соответствует белому цвету, цифра '1' соответствует черному. Все прочие символы (пробелы, переводы строк) должны игнорироваться.

В случае сжатия данных программа должна вывести на стандартный поток вывода (экран) закодированные данные как строка из '0' и '1' и завершить работу

В случае декодирования данных программа должна вывести на стандартный поток вывода (экран) изображение из 51 строк по 100 символов '0' или '1'.

Программа в режиме распаковки данных должна распаковывать данные, сжатой той же самой вашей программой в режиме кодирования данных.

Никаких других данных для распаковки подаваться не будет.

Ваша задача - придумать и реализовать такой алгоритм сжатия и распаковки, который имел бы наилучшую эффективность, то есть размер сжатой картинки. Тестирование будет производиться на наборе тестовом и контрольных наборах. Баллы за каждое изображение будут выставляться в зависимости от степени сжатия

Пример входных данных для кодирования (в условии размер картинки для простоты 3x3):

```
0
010
000
000
```

Возможный пример результата работы:

```
0001
```

Пример входных данных для декодирования:

```
1
0001
```

Результат работы:

```
010
000
000
```

Задача 6: Художник-2

Для создания новой картины мальчик Казимир взял клетчатый лист бумаги размера $4 \times N$. Затем он выбрал произвольную клетку и покрасил её в черный цвет. Каждую следующую клетку он красил так, чтобы у нее была ровно одна

покрашенная соседняя клетка (по сторонам). Так продолжалось до тех пор, пока были клетки которые можно покрасить по этому правилу.

Вы решили повторить работу Казимира и вам требуется оценить число вариантов картин, которые могли получиться.

На стандартном потоке вводится число $N < 10001$

Выведите одно число - количество различных картин, которые могли получиться по модулю 1000000007

Примеры

Входные данные

2

Выходные данные

10

Задача 7: Анти-Бармаглот

Компания *Barmaley's Computing* учла пожелания пользователей и выпустила вторую версию компьютера - *Barmaglot-2*. Поскольку у сотрудников компании несколько странные представления о технологиях, компьютер остался довольно непривычным.

Во-первых, исходные данные он читает с ленты, где могут быть записаны целые числа и латинские буквы. Чтобы показать, что ввод данных закончен, ленту нужно просто оторвать.

Во-вторых, результаты вычислений компьютер печатает на точно такой же ленте. Центральный процессор компьютера оснащён одним регистром; вместо оперативной памяти компьютер оснащён очередью и стеком неограниченного размера. Регистр, а также каждая ячейка очереди и стека могут содержать либо число, либо латинскую букву или пробел, либо специальное значение [BARMALEY], которое используется для обозначения логической лжи, при том что любое другое обозначает истину.

Программа для Barmaglot'a представляет собой строку символов, каждый из которых задаёт машинную команду; программа, состоящая из символов-команд, выполняется последовательно слева направо, кроме двух команд, которые могут нарушить эту последовательность.

- Команды a, b, c и все остальные латинские буквы означают "занести данную букву в регистр".
- Команды 0, 1, ..., 9 означают "занести в регистр соответствующее число".
- Команда @ означает занести в регистр пробел.

- Команды +, -, *, /, % означают соответствующие арифметические действия, операции целочисленные. % - операция взятия остатка
- Команды <, >, = означают сравнение двух чисел или двух символов, & и | означают логическое "и" и логическое "или";

все эти команды используют значение из регистра в качестве левого операнда, значение с вершины стека в качестве правого (оно при этом из стека извлекается), результат заносится обратно в регистр.

- Команда # умножает содержимое регистра в десять раз,
- Команда _ делит содержимое регистра в десять раз.
- Команда ! работает как логическое отрицание содержимого регистра: если там значение [BARMALAY], то заносится значение 1, если любое другое - заносится значение [BARMALAY].
- Команда . выдаёт текущее значение регистра на печать
- команда ? вводит очередное число(целиком, а не по разрядам), а если на вводе кончилась (оборвалась) лента, заносит в регистра значение [BARMALAY].
- Команда] заносит значение из регистра в стек;
- Команда [извлекает значение с вершины стека и заносит его в регистра (если извлекать нечего, в регистр заносится [BARMALAY]);
- Команда ~ меняет местами значения в регистра и на вершине стека.
- Команда } заносит значение из регистра в очередь,
- Команда { извлекает из очереди самое старое значение и помещает в аккумулятор (или помещает туда [BARMALAY], если очередь пуста).
- Команды (и) предназначены для организации ветвлений и циклов и всегда должны в программе стоять парами, то есть в программе должен обязательно соблюдаться баланс круглых скобок. Выполняются они так. Команда (, если в регистре [BARMALAY], идёт по программе вперёд, пока не найдёт парную скобку, и после этого выполнение продолжится со следующей за этой закрывающей скобкой позиции; если в регистре было что-то другое, команда вообще ничего не делает, то есть выполнение продолжается прямо с команды, следующей за ней. Команда), наоборот, если в регистре [BARMALAY], не делает ничего, тогда как если там что-то другое, просматривает программу назад до парной круглой скобки, после чего продолжает выполнение с команды, стоящей после такой скобки справа.

Наконец, команда " прекращает выполнение программы, при этом выполнение считается успешным. Если программа кончилась, не встретив эту команду, она завершается аварийно.

Пробелы в программе игнорируются.

Пример программы, которая печатает традиционную строчку "HELLO WORLD":
H.E.L.L.O.@.W.O.R.L.D."

Вам требуется написать программу которая эмулирует программу на этом языке программирования.

В первой строке вводится программа для Barmaglot. Во второй строке исходное содержимое ленты.

Требуется вывести результат работы программы или "[ERROR]" если выполнение завершилось с ошибкой или программа синтаксически неверная.

Пример работы эмулятора можно найти по ссылке
<http://ejudge.cs.msu.ru/barmaglot2/>

Примеры

Входные данные

```
]?({}?){{(K(~)(>{!})(~)!PЮ)H.@.{}"  
2 10 3 9
```

Выходные данные

```
2 3 9 10
```

Входные данные

```
H.H.  
text
```

Выходные данные

```
[ERROR]
```

Задача 8: Звездчатый многоугольник

Дан многоугольник с вершинами в целочисленных координатах. Будем считать, что точки A и B, лежащие внутри многоугольника, *видны* друг из друга, если отрезок AB лежит внутри многоугольника.

Назовем *ядром* многоугольника множеств точек из которых видны все точки многоугольника. Назовем многоугольник *звездчатым*, если его ядро не пусто.

Требуется написать программу, которая по заданному звездчатому многоугольнику найти его ядро. Можно заметить что ядро звездчатого многоугольника тоже многоугольник

На стандартном потоке ввода задано число вершин звездного многоугольника ($2 < N < 20$). Затем идут N пар целых чисел x_i, y_i - координаты вершин многоугольника в порядке обхода против часовой стрелки. Все координаты не превышают по модулю 10000.

На стандартный поток вывода выведите число вершин ядра исходного многоугольника, затем вершины ядра против часовой стрелки в том же формате, что и во вводе. Среди всех возможных ответов выведите тот, в котором наименьшее число вершин.

Примеры

Входные данные

```
6
0 0
2 0
2 1
1 1
1 2
0 2
```

Выходные данные

```
4
0 0
1 0
1 1
0 1
```