# Олимпиада "Ломоносов" по информатике. 2016-17. 10-11 класс. Второй отборочный тур.

## Задача 1.1: Последовательность

Последовательность целых чисел $\{a_n\}$ задается следующим рекуррентным соотношением:
$a_n = (a_{n-1} + a_{n-2}) \bmod 1000$,
$a_1 = 1$,
$a_2 = 1$,
Где mod - операция взятия остатка от деления.
Все числа последовательности выписали в строку без разделителей.
Отправьте на проверку число с 114 по 187 символ включительно (символы нумеруются с единицы).

## Задача 1.2: Последовательность

Последовательность целых чисел $\{a_n\}$ задается следующим рекуррентным соотношением:
$a_n = (a_{n-1} + a_{n-2}) \bmod 1000$,
$a_1 = 1$,
$a_2 = 1$,
Где mod - операция взятия остатка от деления.
Все числа последовательности выписали в строку без разделителей.
Отправьте на проверку число с 100 по 157 символ включительно (символы нумеруются с единицы).

## Задача 1.3: Последовательность

Последовательность целых чисел $\{a_n\}$ задается следующим рекуррентным соотношением:
$a_n = (a_{n-1} + a_{n-2}) \bmod 1000$,
$a_1 = 1$,
$a_2 = 1$,
Где mod - операция взятия остатка от деления.
Все числа последовательности выписали в строку без разделителей.
Отправьте на проверку число с 124 по 197 символ включительно (символы нумеруются с единицы).

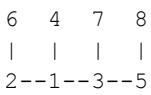
## Задача 1.4: Последовательность

Последовательность целых чисел $\{a_n\}$ задается следующим рекуррентным соотношением:
$a_n = (a_{n-1} + a_{n-2}) \bmod 1000$,
$a_1 = 1$,
$a_2 = 1$,
Где mod - операция взятия остатка от деления.
Все числа последовательности выписали в строку без разделителей.
Отправьте на проверку число с 304 по 407 символ включительно (символы нумеруются с единицы).

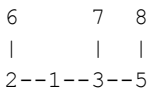
## Задача 2.1: Код Прюфера

Кодирование Прюфера переводит помеченные деревья порядка n в последовательность чисел от 1 до n по алгоритму: Пока количество вершин больше двух:
1) Выбирается лист v с минимальным номером.
2) В код Прюфера добавляется номер вершины, смежной с v.

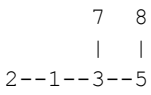
3) Вершина v и инцидентное ей ребро удаляются из дерева.
Полученная последовательность называется кодом Прюфер для заданного дерева.
Рассмотрим построение кода Прюфера на примере. Возьмем дерево:



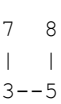
Наименьший лист в дереве 4, номер смежной вершины 1, поэтому в код Прюфера добавляем (1).



На следующем шаге Наименьший лист в дереве 6, номер смежной вершины 2, поэтому в код Прюфера добавляем 2 и получаем (1,2).



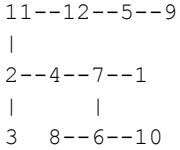
(1,2,1)



(1,2,1,3)

(1,2,1,3,3)

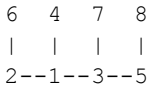
(1,2,1,3,3,5)
(Осталось 2 вершины, завершаем алгоритм. Получаем для примера код Прюфера (1,2,1,3,3,5).
Постройте код Прюфера для дерева:



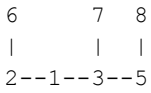
Ответ запишите в таком же виде, как в примере (то есть последовательность чисел записывается в скобках, числа разделяются запятой, пробелов в ответе быть не должно).

## Задача 2.2: Код Прюфера

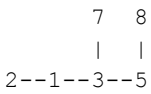
Кодирование Прюфера переводит помеченные деревья порядка n в последовательность чисел от 1 до n по алгоритму: Пока количество вершин больше двух:
1) Выбирается лист v с минимальным номером.
2) В код Прюфера добавляется номер вершины, смежной с v.
3) Вершина v и инцидентное ей ребро удаляются из дерева.
Полученная последовательность называется кодом Прюфер для заданного дерева.
Рассмотрим построение кода Прюфера на примере. Возьмем дерево:



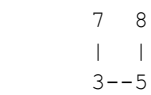

Наименьший лист в дереве 4, номер смежной вершины 1, поэтому в код Прюфера добавляем (1).



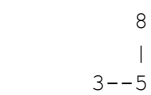

На следующем шаге Наименьший лист в дереве 6, номер смежной вершины 2, поэтому в код Прюфера добавляем 2 и получаем (1,2).




(1,2,1)

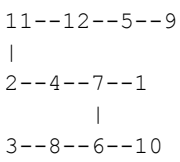



(1,2,1,3)




(1,2,1,3,3)

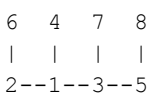
(1,2,1,3,3,5)
(Осталось 2 вершины, завершаем алгоритм. Получаем для примера код Прюфера (1,2,1,3,3,5).
Постройте код Прюфера для дерева:



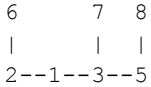
Ответ запишите в таком же виде, как в примере (то есть последовательность чисел записывается в скобках, числа разделяются запятой, пробелов в ответе быть не должно).

## Задача 2.3: Код Прюфера

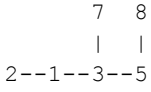
Кодирование Прюфера переводит помеченные деревья порядка n в последовательность чисел от 1 до n по алгоритму: Пока количество вершин больше двух:
1) Выбирается лист v с минимальным номером.
2) В код Прюфера добавляется номер вершины, смежной с v.
3) Вершина v и инцидентное ей ребро удаляются из дерева.
Полученная последовательность называется кодом Прюфер для заданного дерева.
Рассмотрим построение кода Прюфера на примере. Возьмем дерево:



Наименьший лист в дереве 4, номер смежной вершины 1, поэтому в код Прюфера добавляем (1).



На следующем шаге Наименьший лист в дереве 6, номер смежной вершины 2, поэтому в код Прюфера добавляем 2 и получаем (1,2).

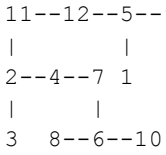


(1,2,1)

(1,2,1,3)

(1,2,1,3,3)

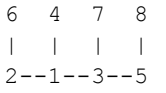
(1,2,1,3,3,5)
(Осталось 2 вершины, завершаем алгоритм. Получаем для примера код Прюфера (1,2,1,3,3,5).
Постройте код Прюфера для дерева:



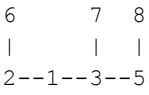
Ответ запишите в таком же виде, как в примере (то есть последовательность чисел записывается в скобках, числа разделяются запятой, пробелов в ответе быть не должно).

## Задача 2.4: Код Прюфера

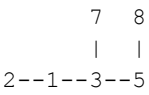
Кодирование Прюфера переводит помеченные деревья порядка n в последовательность чисел от 1 до n по алгоритму: Пока количество вершин больше двух:
1) Выбирается лист v с минимальным номером.
2) В код Прюфера добавляется номер вершины, смежной с v.
3) Вершина v и инцидентное ей ребро удаляются из дерева.
Полученная последовательность называется кодом Прюфер для заданного дерева.
Рассмотрим построение кода Прюфера на примере. Возьмем дерево:



Наименьший лист в дереве 4, номер смежной вершины 1, поэтому в код Прюфера добавляем (1).



На следующем шаге Наименьший лист в дереве 6, номер смежной вершины 2, поэтому в код Прюфера добавляем 2 и получаем (1,2).

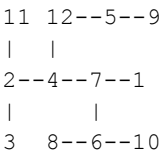


(1,2,1)

(1,2,1,3)

(1,2,1,3,3)

(1,2,1,3,3,5)
(Осталось 2 вершины, завершаем алгоритм. Получаем для примера код Прюфера (1,2,1,3,3,5).
Постройте код Прюфера для дерева:

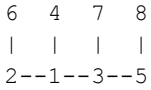


Ответ запишите в таком же виде, как в примере (то есть последовательность чисел записывается в скобках, числа разделяются запятой, пробелов в ответе быть не должно).

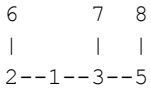
# Задача 3: Код Прюфера - 2

Кодирование Прюфера переводит помеченные деревья порядка n в последовательность чисел от 1 до n по алгоритму: Пока количество вершин больше двух:
1) Выбирается лист v с минимальным номером.

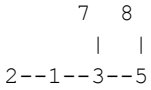
2) В код Прюфера добавляется номер вершины, смежной с v.
3) Вершина v и инцидентное ей ребро удаляются из дерева.
Полученная последовательность называется кодом Прюфер для заданного дерева.
Рассмотрим построение кода Прюфера на примере. Возьмем дерево:



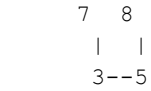
Наименьший лист в дереве 4, номер смежной вершины 1, поэтому в код Прюфера добавляем (1).



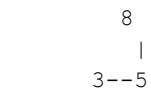
На следующем шаге Наименьший лист в дереве 6, номер смежной вершины 2, поэтому в код Прюфера добавляем 2 и получаем (1,2).



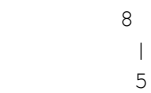
(1,2,1)



(1,2,1,3)



(1,2,1,3,3)



(1,2,1,3,3,5)
(Осталось 2 вершины, завершаем алгоритм. Получаем для примера код Прюфера (1,2,1,3,3,5).
Напишите программу, которая для заданного кода Прюфера из n-2 чисел от 1 до n восстанавливает дерево
На ввод подается N-2 целых чисел в диапазоне от 1 до N.
Выведите все ребра дерева как пары вершин, которые являются концами ребра. Вершины в ребре необходимо упорядочить по возрастанию, список ребер отсортируйте.

## Примеры

**Входные данные**

1 2 1 3 3 5

**Выходные данные**

1 2
1 3
1 4
2 6
3 5
3 7
5 8

# Задача 4: Последовательность - 2

Последовательность целых чисел $\{a_n\}$ задается следующим рекуррентным соотношением:
$a_n = (a_{n-1} + a_{n-2}) \bmod K$,
$a_1 = A$,
$a_2 = B$,
Где mod - операция взятия остатка деления.
Если выписать всю последовательность, то в какой-то момент числа в ней начнут повторяться. Ваша задача для заданных A, B, K ($0 < A, B, K < 1000$) определить длину цикла в последовательности.
Длиной цикла c называется такое минимальное число $p > 0$, что существует некоторый начальный номер элемента последовательности n0, что для любого $n \geq n0$ выполняется $a_n = a_{n+p}$.
На стандартный поток вводятся три числа A, B и K. Ответ выводится на стандартный поток вывода. Числа на стандартном потоке ввода разделяются произвольным количеством пробельных символов (переводов строк, пробелов и табуляций).

## Примеры

**Входные данные**

1 1 10

**Выходные данные**

60

# Задача 5: Сжатие текста

Важный раздел информатики - это алгоритмы сжатия данных. В этой задаче мы будем рассматривать только сжатие английских текстов без потерь. Задача заключается в том, что входной текст 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 *естественных* текстов на английском языке.
На вход программе подается либо текст 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 сжатия далее на вход подается текст состоящий из символов с кодами ascii до 127.
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 исходный текст.
Программа в режиме распаковки данных должна распаковывать данные, сжатой той же самой вашей программой в режиме закодирования данных. Никаких других данных для распаковки подаваться не будет.
Ваша задача - придумать и реализовать такой алгоритм сжвтия и распаковки, который имел бы наилучшую эффективность, то есть размер сжатого текста.
Тестирование будет производиться на наборе тестовом и контрольных наборах. Баллы за каждый текст будут выставляться в зависимости от степени сжатия
Пример входных данных для кодирования

```
0
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
```

Возможный пример результата работы:

```
01100101 01001110 01110000 01111010 01111010 01000011 01110011 01110101
01010100 01111001 00110001 01010011 01001011 01001101 01101100 01011000
01001011 01001101 01101100 01001001 01010110 01010001 01101010 01001110
01001011 01100011 01101110 01001101 01010100 01010011 01111000 01001010
01010110 01010001 01100111 01110011 01010100 01010011 00110000 01110101
01111001 01100011 01111010 01010000 01010101 00111000 01101000 01010000
01010101 00101111 01000100 01001010 01010100 01000101 01110110 01010110
01100111 01010101 01101010 01101110 01011010 01011010 01100001 01101100
01000110 01101000 01010101 01000100 01100101 01011001 01101100 00110101
01001011 01010001 01110001 01110101 01010001 01001000 01011010 01101100
01010011 01010101 01011010 01101101 01011000 01101010 01101111 01110110
01101100 00101011 01001111 01101111 01001101 01100001 01010000 01000111
01000100 01001010 01010001 01111000 01000001 01001100 01010000 01101111
00101011 01111001 01000001 00111101
```

Пример входных данных для декодирования:

1
01100101010011100111000001111010011110100100001101110011011101010101010000111
10010011000101010011010010110100110101101100010110000100101101001101011011 00
01001001010101100101000101101010010011100100101101100011011011100100110101 01
01000101001101111000010010100101011001010001011001110111001101010100010100 11
00110000011101010111100101100011011110100101000001010101001110000110100001 01
00000101010100101111010001000100101001010100010001010111011001010110011001 11
01010101011010100110111001011010010110100110000101101100010001100110100001 01
01010100010001100101010110010110110000110101010010110101000101110001011101 01
01010001010010000101101001101100010100110101010101011010011011010101100001 10
10100110111101110110011011000010101101001111011011110100110101100001010100 00
01000111010001000100101001010001011110000100000101001100010100000110111100 10
10110111100101000001001111 01

Результат работы:

Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything
Answer to the Ultimate Question of Life, the Universe, and Everything

# Задача 6: Художник-3

Для создания новой картины мальчик Казимир взял клечатый лист бумаги размера 6*N. Затем он выбрал произвольную клетку и покрасил её в черный цвет. Каждую следующую клетку он красил так, чтобы она соприкасалась ровно в одной точке ровно с одной из уже покрашенных. Так продолжалось до тех пор, пока были клетки котые 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 < 101
Выведите одно число - количество различных картин, которые могли получиться по модулю 1000000007

## Примеры

**Входные данные**

2

**Выходные данные**

2

# Задача 7: Befunge

Befunge — стековый эзотерический язык программирования. Считается двумерным, так как программа на Befunge записывается в таблицу со сшитыми краями (тор), по которой в различных направлениях перемещается интерпретатор, исполняя команды, расположенные в её ячейках. Размер таблицы ограничен 25 строками и 80 столбцами.
Ниже перечислены команды языка Befunge. Каждая команда кодируется одним ASCII-символом. Команды, берущие параметры из стека, удаляют их со стека.

перемещение (9):
> Двигаться вправо
< Двигаться влево
^ Двигаться вверх
v Двигаться вниз
_ Двигаться вправо, если на вершине стека 0, иначе — влево.
| Двигаться вниз, если на вершине стека 0, иначе — вверх.
? Двигаться в случайном направлении
# Пропустить следующую ячейку ("трамплин")
@ Конец программы
манипулирование со стеком (3):
: Поместить в стек копию вершины
\ Обменять местами вершину и подвершину
$ Удалить вершину
модификация кода программы (2):
p "PUT": со стека извлекаются координаты ячейки и ASCII-код символа, который помещается по этим координатам
g "GET": со стека извлекаются координаты ячейки; ASCII-код символа по этим координатам помещается в стек
константы (2):
0-9 Поместить число в стек
" Начало/конец символьного режима, в котором ASCII-коды всех текущих символов программы помещаются в стек
стековые арифметические операции (5):
+ Сложение вершины и подвершины
- Вычитание вершины и подвершины
* Умножение вершины и подвершины
/ Целочисленное деление
% Остаток от деления
стековые логические операции (2):
! Отрицание: нуль на вершине заменяется на 1, ненулевое значение — на 0
` Сравнение "больше, чем": если подвершина больше вершины, в стек помещается 1, иначе 0 (forth:>)
ввод-вывод (4):
& Запросить у пользователя число и поместить его в стек
~ Запросить у пользователя символ и поместить в стек его ASCII-код
. Распечатать вершину стека как целое число
, Распечатать символ, соответствующий ASCII-коду на вершине стека

Вам требуется написать интерпретатор этого языка прогрммирования.
В первой строке задается строка с входными данными программы. В последующих строках программа на языке Befunge.
Выведите результат работы программы. Считайте, что программа не зацикливается.

## Примеры

**Входные данные**

```
1
62*1+v>01p001>+v>\:02p\:02gv
     0       ^             <
     .          :p
     "          .1
        v 0," "<0
     "  >1g12-+:|
     ,           @
     >^
```

**Выходные данные**

0 1 1 2 3 5 8 13 21 34 55 89 144 233

# 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* многоугольлника множесто точек из которых видны все точки многоугольника. Назовем многоугольник *звёздчатым*, если его ядро не пусто.

Требуется написать программу, которая по заданному набору вершин строит звездчатый многоугольник состоящий из этих вершин

На стандартном потоке ввода задано число вершин множества ($2 < N < 1000000$). Затем идут N пар целых чисел $x_i$, $y_i$ - координаты точек. Все координаты не превышают по модулю 10000.
На стандартный поток вывода выведите вершины многоугольника в порядке обхода против часовой стрелки.

## Примеры

**Входные данные**

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

**Выходные данные**

0 0
2 0
2 1
1 1
1 2
0 2