VK Cup 2021 - Квалификация (Engine)

Statement is not available on English language

A. Зингер | color

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Немногие знают, что сотрудники ВКонтакте могут менять цвет подсветки в куполе знаменитого Дома Зингера, где расположена штаб-квартира ВКонтакте. Для этого нужно всего лишь отправить сообщение с цветом в специальный чат «Зингер | color», а бот его распознает и сменит подсветку. При этом на время городских мероприятий смена цвета блокируется.

Формально, бот обрабатывает три типа сообщений:

- `lock`: заблокировать изменение цвета. Если оно и так заблокировано на данный момент, сообщение игнорируется.
- `unlock`: разблокировать изменение цвета. Если оно и так разблокировано на данный момент, сообщение игнорируется.
- `red` / `orange` / `yellow` / `green` / `blue` / `indigo` / `violet`: изменить цвет купола на заданный, если изменение цвета на данный момент не заблокировано.

Вам дана история сообщений, полученных ботом, в хронологическом порядке. Считайте, что перед получением первого сообщения купол подсвечивается голубым (`blue`), а изменение цвета не заблокировано.

Определите, какой цвет будет у купола Дома Зингера после обработки этих сообщений.

Входные данные
В первой строке задано одно целое число n ($1 \le n \le 100$) — число сообщений, полученных ботом.

В следующих n строках заданы сообщения, полученные ботом, в хронологическом порядке, по одному сообщению в строке. Каждое сообщение — строка из следующего набора: `lock`, `unlock`, `red`, `orange`, `yellow`, `green`, `blue`, `indigo`, `violet`.

Выходные данные
Выведите цвет купола после обработки сообщений ботом.

Примеры

входные данные
7 red violet unlock red orange lock indigo
выходные данные
orange

входные данные
5 lock unlock lock unlock unlock
выходные данные
blue

Statement is not available on English language

B1. Мониторинг

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт

ввод: стандартный ввод
вывод: стандартный вывод

ВКонтакте открыла второй штаб в Санкт-Петербурге! Вы не преминули возможностью сменить обстановку и решили переехать из офиса в Доме Зингера в офис на Красном мосту.

Для комфортной работы вам потребуются два монитора с одинаковой высотой, чтобы изображение на них выглядело единым целым. На складе офиса на Красном мосту есть n мониторов, i-й из них имеет ширину w_i и высоту h_i. Любой монитор можно повернуть на 90 градусов, и тогда он будет иметь ширину h_i и высоту w_i.

Назовём неупорядоченную пару из двух различных мониторов подходящей, если можно их повернуть так, чтобы они имели одинаковую высоту. Любой из мониторов в паре можно как повернуть относительно исходной ориентации, так и не поворачивать.

Подсчитайте подходящие пары мониторов.

Входные данные
В первой строке задано одно целое число n — число мониторов на складе.

В каждой из следующих n строк заданы два целых числа w_i и h_i ($1 \le w_i, h_i \le 10^9$) — ширина и высота i-го монитора. Обратите внимание, что мониторы могут быть квадратными ($w_i = h_i$), а размеры разных мониторов могут совпадать.

В этой версии задачи $2 \le n \le 10^3$.

Выходные данные
Выведите число подходящих пар мониторов.

Примеры

входные данные
5 3 2 2 2 5 5 3 5 4 3
выходные данные
5

входные данные
7 10 10 10 20 20 10 10 20 10 20 10 10 20 10
выходные данные
21

Примечание
В первом примере подходящими являются пары мониторов с номерами $(1, 2)$, $(1, 4)$, $(1, 5)$, $(3, 4)$, $(4, 5)$.

Во втором примере все пары мониторов — подходящие.

Statement is not available on English language

B2. Мониторинг

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

ВКонтакте открыла второй штаб в Санкт-Петербурге! Вы не преминули возможностью сменить обстановку и решили переехать из офиса в Доме Зингера в офис на Красном мосту.

Для комфортной работы вам потребуются два монитора с одинаковой высотой, чтобы изображение на них выглядело единым целым. На складе офиса на Красном мосту есть n мониторов, i-й из них имеет ширину w_i и высоту h_i. Любой монитор можно повернуть на 90 градусов, и тогда он будет иметь ширину h_i и высоту w_i.

Назовём неупорядоченную пару из двух различных мониторов подходящей, если можно их повернуть так, чтобы они имели одинаковую высоту. Любой из мониторов в паре можно как повернуть относительно исходной ориентации, так и не поворачивать.

Подсчитайте подходящие пары мониторов.

Входные данные
В первой строке задано одно целое число n — число мониторов на складе.

В каждой из следующих n строк заданы два целых числа w_i и h_i ($1 \le w_i, h_i \le 10^9$) — ширина и высота i-го монитора. Обратите внимание, что мониторы могут быть квадратными ($w_i = h_i$), а размеры разных мониторов могут совпадать.

В этой версии задачи $2 \le n \le 10^5$.

Выходные данные
Выведите число подходящих пар мониторов.

Примеры

входные данные
5 3 2 2 2 5 5 3 5 4 3
выходные данные
5

входные данные
7 10 10 10 20 20 10 10 20 10 20 10 10 20 10
выходные данные
21

Примечание
В первом примере подходящими являются пары мониторов с номерами $(1, 2)$, $(1, 4)$, $(1, 5)$, $(3, 4)$, $(4, 5)$.

Во втором примере все пары мониторов — подходящие.

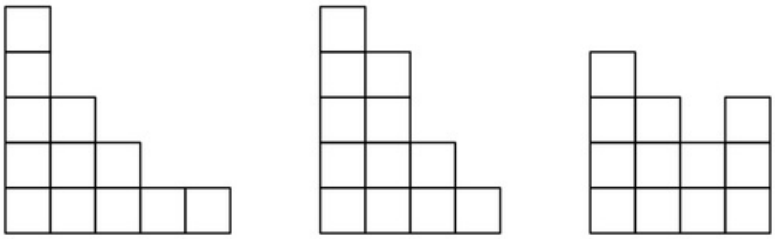
Statement is not available on English language

C. Симметричный амфитеатр

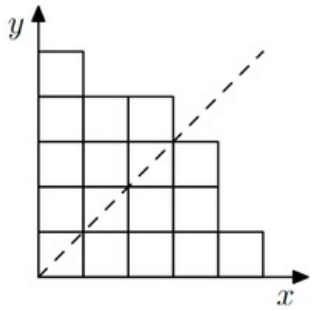
ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Все готовятся к VK Fest 2021! Для того, чтобы зрителям была лучше видна главная сцена, планируется построить амфитеатр. В этой задаче мы будем рассматривать его сбоку — схематично он будет иметь форму лестницы из n одинаковых квадратов. Лестница — это одна или более башен квадратов, выстроенных в ряд, где высоты башен невозрастают слева направо.

На следующем рисунке можно видеть три разные фигуры из 12 квадратов. Первые две фигуры — лестницы, а третья — нет.



Из эстетических соображений было решено, что амфитеатр должен быть симметричным. Формально, амфитеатр называется симметричным, если при отражении его схемы относительно прямой $x = y$ получается тот же самый рисунок (где ось x направлена слева направо, а ось y — снизу вверх). Например, первая лестница на рисунке выше — симметричная, а вторая — нет.



Кроме того, амфитеатр должен быть максимально компактным — а именно, сторона минимального квадрата, внутрь которого можно его поместить, должна быть как можно меньше.

По заданному числу n нарисуйте схему амфитеатра из ровно n квадратов, удовлетворяющую всем условиям.

Входные данные
В единственной строке задано одно целое число n ($1 \le n \le 100$) — число квадратов, из которых нужно составить схему амфитеатра.

Выходные данные
Если не существует схемы амфитеатра из n квадратов, выведите единственное число -1.

Иначе в первой строке выведите целое число m — минимальное возможное число строк и столбцов в схеме амфитеатра. Далее выведите m строк, описывающих схему. Каждая строка должна содержать ровно m символов 'o' (строчная латинская буква) или '`.`', где 'o' описывает построенный квадрат, а '`.`' — пустое место. Схема амфитеатра должна состоять ровно из n символов 'o'. Ячейка в левом нижнем углу должна содержать квадрат. Если возможных ответов с минимальным m несколько, выведите любой из них.

Примеры

входные данные
3
выходные данные
2 o. oo

входные данные
17
выходные данные
5 o.... ooo.. oooo. oooo. ooooo

Statement is not available on English language

D. Редактируем Зингер | color

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Немногие знают, что сотрудники ВКонтакте могут менять цвет подсветки в куполе знаменитого Дома Зингера, где расположена штаб-квартира ВКонтакте. Для этого нужно всего лишь отправить сообщение с цветом в специальный чат «Зингер | color», а бот его распознает и сменит подсветку. При этом на время городских мероприятий смена цвета блокируется.

Формально, бот обрабатывает три типа сообщений:

- `lock`: заблокировать изменение цвета. Если оно и так заблокировано на данный момент, сообщение игнорируется.
- `unlock`: разблокировать изменение цвета. Если оно и так разблокировано на данный момент, сообщение игнорируется.
- `red` / `orange` / `yellow` / `green` / `blue` / `indigo` / `violet`: изменить цвет купола на заданный, если изменение цвета на данный момент не заблокировано.

Вам дана история сообщений, полученных ботом, в хронологическом порядке. Считайте, что перед получением первого сообщения купол подсвечивается голубым (`blue`), а изменение цвета не заблокировано.

В качестве эксперимента было решено поддержать в боте эффективную обработку редактирования сообщений. Вам дана последовательность пар вида (i, msg), означающих, что i-е в хронологическом порядке сообщение было отредактировано

и теперь имеет вид msg. Обратите внимание, что редактироваться может любое сообщение, и при редактировании сообщения бот должен обработать всю историю сообщений заново (в частности, перед обработкой первого сообщения цвет купола голубой, а изменение цвета не заблокировано).

Определите, какой цвет будет у купола Дома Зингера до первой операции редактирования, а также после каждой операции редактирования.

Входные данные

В первой строке задано одно целое число n ($1 \le n \le 10^5$) — число сообщений, полученных ботом.

В следующих n строках заданы сообщения, полученные ботом, в хронологическом порядке, по одному сообщению в строке. Каждое сообщение — строка из следующего набора: `lock`, `unlock`, `red`, `orange`, `yellow`, `green`, `blue`, `indigo`, `violet`. Сообщения пронумерованы от 1 до n.

В следующей строке задано одно целое число t ($1 \le t \le 10^5$) — число операций редактирования сообщений.

В следующих t строках заданы операции редактирования в хронологическом порядке, по одной в строке. Каждая операция — пара из номера сообщения i ($1 \le i \le n$) и его нового содержимого msg, также принадлежащего набору `lock`, `unlock`, `red`, `orange`, `yellow`, `green`, `blue`, `indigo`, `violet`.

Выходные данные

Выведите $t + 1$ строку: цвет купола до первой операции редактирования, а также после каждой операции редактирования в хронологическом порядке.

Примеры

входные данные
7 red violet unlock red orange lock indigo 6 5 green 6 lock 6 yellow 4 lock 1 lock 5 unlock

выходные данные
orange green green indigo violet blue indigo

входные данные
1 red 8 1 lock 1 unlock 1 blue 1 unlock 1 unlock 1 lock 1 yellow 1 lock

выходные данные
red blue blue blue blue blue blue yellow blue

Statement is not available on English language

E1. Сортировка слиянием

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт

ввод: стандартный ввод
вывод: стандартный вывод

Рассмотрим следующий код сортировки слиянием на языке Python:

```
def sort(a):
  n = len(a)
  b = [0 for i in range(n)]
  log = []

  def mergeSort(l, r):
    if r - l <= 1:
      return
    m = (l + r) >> 1
    mergeSort(l, m)
    mergeSort(m, r)
    i, j, k = l, m, l
    while i < m and j < r:
      if a[i] < a[j]:
        log.append('0')
        b[k] = a[i]
        i += 1
      else:
        log.append('1')
        b[k] = a[j]
        j += 1
      k += 1
    while i < m:
      b[k] = a[i]
      i += 1
      k += 1
    while j < r:
      b[k] = a[j]
      j += 1
      k += 1
    for p in range(l, r):
      a[p] = b[p]

  mergeSort(0, n)
  return "".join(log)
```

Как можно заметить, этот код использует логирование — важнейший инструмент разработки.

Старший разработчик ВКонтакте Вася сгенерировал перестановку a (массив из n различных целых чисел от 1 до n), дал её на вход функции `sort` и получил на выходе строку s. На следующий день строку s Вася нашёл, а перестановка a потерялась.

Вася хочет восстановить любую перестановку a такую, что вызов функции `sort` от неё даст ту же строку s. Помогите ему!

Входные данные

Ввод содержит непустую строку s, состоящую из символов `0` и `1`.

В этой версии задачи для любого теста существует перестановка длины 16, удовлетворяющая условию. Тем не менее, ваш ответ может иметь любую длину, в том числе отличную от 16.

Выходные данные

В первой строке выведите целое число n — длину перестановки.

Во второй строке выведите n различных целых чисел $a_0, a_1, \ldots, a_{n-1}$ ($1 \le a_i \le n$) — элементы перестановки.

Если существует несколько вариантов ответа, выведите любой из них.

Примеры

входные данные
00000000000000000000000000000000
выходные данные
16 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

входные данные
11111111111111111111111111111111

выходные данные
16 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

входные данные
1010110100011001000110110011110110000111110010
выходные данные
16 13 6 1 7 12 5 4 15 14 16 10 11 3 8 9 2

Statement is not available on English language

E2. Сортировка слиянием

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Рассмотрим следующий код сортировки слиянием на языке Python:

```
def sort(a):
  n = len(a)
  b = [0 for i in range(n)]
  log = []

  def mergeSort(l, r):
    if r - l <= 1:
      return
    m = (l + r) >> 1
    mergeSort(l, m)
    mergeSort(m, r)
    i, j, k = l, m, l
    while i < m and j < r:
      if a[i] < a[j]:
        log.append('0')
        b[k] = a[i]
        i += 1
      else:
        log.append('1')
        b[k] = a[j]
        j += 1
      k += 1
    while i < m:
      b[k] = a[i]
      i += 1
      k += 1
    while j < r:
      b[k] = a[j]
      j += 1
      k += 1
    for p in range(l, r):
      a[p] = b[p]

  mergeSort(0, n)
  return "".join(log)
```

Как можно заметить, этот код использует логирование — важнейший инструмент разработки.

Старший разработчик ВКонтакте Вася сгенерировал перестановку a (массив из n различных целых чисел от 1 до n), дал её на вход функции `sort` и получил на выходе строку s. На следующий день строку s Вася нашёл, а перестановка a потерялась.

Вася хочет восстановить любую перестановку a такую, что вызов функции `sort` от неё даст ту же строку s. Помогите ему!

Входные данные

Ввод содержит непустую строку s, состоящую из символов `0` и `1`.

В этой версии задачи для любого теста существует перестановка длины не более 10^3, удовлетворяющая условию. Тем не

менее, ваш ответ может иметь любую длину, в том числе превышающую 10^3.

Выходные данные

В первой строке выведите целое число n — длину перестановки.

Во второй строке выведите n различных целых чисел $a_0, a_1, \ldots, a_{n-1}$ ($1 \leq a_i \leq n$) — элементы перестановки.

Если существует несколько вариантов ответа, выведите любой из них.

Примеры

входные данные
00000000000000000000000000000000
выходные данные
16 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

входные данные
11111111111111111111111111111111
выходные данные
16 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

входные данные
10101101000110010001101100111101100001111 0010
выходные данные
16 13 6 1 7 12 5 4 15 14 16 10 11 3 8 9 2

Statement is not available on English language

E3. Сортировка слиянием

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 512 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Рассмотрим следующий код сортировки слиянием на языке Python:

```
def sort(a):
  n = len(a)
  b = [0 for i in range(n)]
  log = []

  def mergeSort(l, r):
    if r - l <= 1:
      return
    m = (l + r) >> 1
    mergeSort(l, m)
    mergeSort(m, r)
    i, j, k = l, m, l
    while i < m and j < r:
      if a[i] < a[j]:
        log.append('0')
        b[k] = a[i]
        i += 1
      else:
        log.append('1')
        b[k] = a[j]
        j += 1
      k += 1
    while i < m:
      b[k] = a[i]
      i += 1
      k += 1
    while j < r:
      b[k] = a[j]
```

```
    j += 1
    k += 1
  for p in range(l, r):
    a[p] = b[p]

 mergeSort(0, n)
 return "".join(log)
```

Как можно заметить, этот код использует логирование — важнейший инструмент разработки.

Старший разработчик ВКонтакте Вася сгенерировал перестановку a (массив из n различных целых чисел от 1 до n), дал её на вход функции `sort` и получил на выходе строку s. На следующий день строку s Вася нашёл, а перестановка a потерялась.

Вася хочет восстановить любую перестановку a такую, что вызов функции `sort` от неё даст ту же строку s. Помогите ему!

Входные данные

Ввод содержит непустую строку s, состоящую из символов `0` и `1`.

В этой версии задачи для любого теста существует перестановка длины не более 10^5, удовлетворяющая условию. Тем не менее, ваш ответ может иметь любую длину, в том числе превышающую 10^5.

Выходные данные

В первой строке выведите целое число n — длину перестановки.

Во второй строке выведите n различных целых чисел $a_0, a_1, \ldots, a_{n-1}$ ($1 \le a_i \le n$) — элементы перестановки.

Если существует несколько вариантов ответа, выведите любой из них.

Примеры

входные данные
00000000000000000000000000000000
выходные данные
16 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

входные данные
11111111111111111111111111111111
выходные данные
16 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

входные данные
10101101000110010001101100111101100001111 0010
выходные данные
16 13 6 1 7 12 5 4 15 14 16 10 11 3 8 9 2