

2. (7)

Одной из основных функций операционной системы является эффективное распределение ресурсов между процессами. Рассмотрим модель, в которой под управлением операционной системы работает два процесса: p1 и p2. Каждый процесс попеременно реализует некоторое количество операций вычисления на процессоре и некоторое количество операций ввода-вывода. В системе один процессор, который в один момент времени может исполнять вычислительные операции только одного процесса. Подсистема ввода-вывода может независимо и параллельно обрабатывать операции ввода-вывода обоих процессов без изменения производительности.

Время модели дискретно и измеряется в условных тактах. Для каждого процесса заданы в этих условных тактах два параметра: CPU-burst и I/O-burst – время непрерывного выполнения вычислительных операций и время непрерывного выполнения операций ввода-вывода.

Процесс	CPU-burst	I/O-burst
p1	41	73
p2	52	84

Для управления распределением процессорного времени используется алгоритм Round Robin. Каждому процессу поочередно дается квант непрерывного выполнения на процессоре, равный 20 тактам. Реализация алгоритма отвечает следующим правилам:

1. Процессам поочередно предоставляется возможность непрерывного использования процессора, продолжительностью, не превышающей квант непрерывного исполнения. В первый такт такая возможность предоставляется процессору p1. При этом процесс p2 также готов исполняться на процессоре.
2. Если процесс использовал целиком квант непрерывного исполнения (возможно, полностью исчерпав очередной CPU-burst), он вытесняется и процессор передается другому процессу. На переключение между процессами тратится фиксированное время – 5 тактов.
3. Если процесс, не использовав целиком квант непрерывного исполнения полностью, исчерпывает свой CPU-burst, он незамедлительно начинает операции ввода-вывода, продолжающиеся количество тактов, равное I/O-burst этого процесса. Одновременно с этим начинается переключение на другой процесс, которое также длится 5 тактов.
4. Если после переключения процессов у очередного процесса есть остаток CPU-burst, в т.ч. он только что завершил очередной цикл операций ввода-вывода, процесс выполняется в соответствии с пунктами 2 или 3.
5. Если после переключения процессов оказывается, что у очередного процесса не завершился цикл операций ввода-вывода, незамедлительно запускается переключение на другой процесс, которое длится также 5 тактов.
6. Операции ввода-вывода не прерываются и могут выполняться параллельно с вычислениями другого процесса или операциями переключения процессов.

Определите для каждого из процессов суммарное количество тактов, которое этот процесс вычислялся на процессоре (время на операции ввода вывода и на переключения процессов в него не входит, только фактическое использование процессора данным процессом для вычисления) после 500 тактов от начала работы модели. В ответе укажите через пробел два целых числа: значение для процесса p1 и затем значение для процесса p2.

13	1. (13)
?	2. (7)
?	3. (9)
?	4. (11)
?	5. (8)
?	6. (7)
7	7. (7)
8	8. (8)
×	9. (13)
17	10. (17)

7. (7)

Ограничение времени	1 секунда
Ограничение памяти	512Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В этой задаче вам необходимо написать прототип фрагмента операционной системы для мобильных телефонов.

На вход подается два события: начало касания и конец касания. Каждое событие характеризуется временем в миллисекундах и координатами на экране.

На выход вам необходимо выдать одно из четырех действий, которое совершил пользователь:

1. Случайное касание. Если время между событиями меньше 100 миллисекунд, то такое касание считается случайным.
2. Короткое нажатие. Если время между событиями не меньше 100 и не больше 1000 миллисекунд, а расстояние между начальной и конечной точкой не больше 50 единиц, то такое касание считается коротким нажатием. Расстояние между начальной (x_1, y_1) и конечной (x_2, y_2) точкой считается как $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$.
3. Длинное нажатие. Если время между событиями больше 1000 миллисекунд, а расстояние между начальной и конечной точкой не больше 50 единиц, то такое касание считается длинным нажатием.
4. Перетаскивание. Если время между событиями не меньше 100 и расстояние между начальной и конечной точкой больше 50 единиц, то такое касание считается перетаскиванием.

По заданным параметрам событий начала и конца касания определите, какой тип действия совершил пользователь.

Формат ввода

Входные данные содержат один или несколько тестовых примеров. На первой строке ввода находится число C — количество тестовых примеров ($1 \leq C \leq 1000$).

Далее следуют C пар строк, описывающих тестовые примеры.

В первой строке описания находятся три целых числа t_1 , x_1 и y_1 — время и координаты начала касания.

Во второй строке описания находятся три целых числа t_2 , x_2 и y_2 — время и координаты конца касания.

Время задается в миллисекундах.

Гарантируется, что $0 \leq t_1 < t_2 \leq 10^9$, $0 \leq x_1, y_1, x_2, y_2 \leq 8192$.

Формат вывода

Для каждого тестового примера выведите целое число от 1 до 4 — тип действия пользователя, согласно описанию в условии задачи.

13	1. (13)
?	2. (7)
?	3. (9)
?	4. (11)
?	5. (8)
?	6. (7)
7	7. (7)
8	8. (8)
×	9. (13)
17	10. (17)

Формат ввода

Входные данные содержат один или несколько тестовых примеров. На первой строке ввода находится число C — количество тестовых примеров ($1 \leq C \leq 1000$).

Далее следуют C пар строк, описывающих тестовые примеры.

В первой строке описания находятся три целых числа t_1 , x_1 и y_1 — время и координаты начала касания.

Во второй строке описания находятся три целых числа t_2 , x_2 и y_2 — время и координаты конца касания.



Время задается в миллисекундах.

Гарантируется, что $0 \leq t_1 < t_2 \leq 10^9$, $0 \leq x_1, y_1, x_2, y_2 \leq 8192$.

Формат вывода

Для каждого тестового примера выведите целое число от 1 до 4 — тип действия пользователя, согласно описанию в условии задачи.

Пример

Ввод 	Вывод 
4	1
0 200 200	2
20 201 203	3
0 200 200	4
200 193 212	
0 200 200	
2000 201 199	
0 200 200	
805 913 324	

Ограничение времени	1 секунда
Ограничение памяти	512Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

- ?

3. (9)
- ?

4. (11)
- ?

5. (8)
- ?

6. (7)
- 7

7. (7)
- 8

8. (8)
- ✖

9. (13)
- 17

10. (17)

Задан массив положительных целых чисел и число b . Назовем b -весом числа сумму квадратов его цифр в системе счисления с основанием b .

Назовем массив b -отсортированным, если его элементы упорядочены по возрастанию b -веса.

Требуется переставить элементы заданного массива в таком порядке, чтобы он стал b -отсортированным. Числа, которые имеют одинаковый b -вес, должны остаться в том же порядке, в котором они шли в исходном массиве.

Формат ввода

Первая строка ввода содержит два целых числа: n — длину массива, и b — систему счисления, в которой вычисляется вес элементов ($1 \leq n \leq 100\,000$, $2 \leq b \leq 100$).

На второй строке заданы n положительных целых чисел — элементы массива. Элементы массива не превосходят 10^9 .

Формат вывода

Выведите результат сортировки массива по b -весу его элементов.

Пример

Ввод	<div>8 2</div> <div>3 1 4 1 5 7 2 6</div>	Вывод	<div>1 4 1 2 3 5 6 7</div>
------	---	-------	----------------------------

Примечания

В двоичной системе счисления поданный на вход в примере массив выглядит так: $[11_2, 1_2, 100_2, 1_2, 101_2, 111_2, 10_2, 110_2]$. Веса его элементов: $w(11_2) = 1^2 + 1^2 = 2$, $w(1_2) = 1^2 = 1$, $w(100_2) = 1$, $w(1_2) = 1$, $w(101_2) = 2$, $w(111_2) = 3$, $w(10_2) = 1$, $w(110_2) = 2$.

Ограничение времени	1 секунда
Ограничение памяти	512Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Задана строка s длиной n из нулей и единиц: $s = s_1s_2 \dots s_n$.

Последовательность индексов $1 \leq i_1 < i_2 < \dots < i_k \leq n$ задает подпоследовательность заданной строки $s_{i_1}s_{i_2} \dots s_{i_k}$. Будем называть последовательность индексов *интересной*, если соответствующая строка имеет два соседних нуля. Иначе говоря, должны найтись два таких соседних индекса i_j и i_{j+1} , что $s_{i_j} = s_{i_{j+1}} = 0$.

Требуется найти количество различных интересных последовательностей индексов и вывести остаток от деления этого количества на $10^9 + 7$.

Формат ввода

Входные данные содержат один или несколько тестовых примеров. На первой строке ввода находится число C — количество тестовых примеров ($1 \leq C \leq 1000$).

Далее следуют C строк, задающих тестовые примеры.

Все строки состоят из нулей и единиц, длина каждой строки от 1 до 10^5 , суммарная длина строк во всех тестовых примерах одних входных данных не превышает 10^5 .

Формат вывода

Для каждого тестового примера выведите одно число — количество интересных последовательностей индексов.

Пример

Ввод	Вывод
4	4
000	2
001	0
1111	1
010	

?	3. (9)
?	4. (11)
?	5. (8)
?	6. (7)
7	7. (7)
8	8. (8)
×	9. (13)
17	10. (17)

Ограничение времени	3 секунды
Ограничение памяти	512Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

?	3. (9)
?	4. (11)
?	5. (8)
?	6. (7)
7	7. (7)
8	8. (8)
×	9. (13)
17	10. (17)

Компьютерную сеть можно рассмотреть как n серверов, пронумерованных от 1 до n . Некоторые пары серверов напрямую соединены двусторонними каналами связи. Известно, что можно передать сообщение от любого сервера до любого другого с использованием каналов связи, возможно используя промежуточные серверы.

Каналы связи могут выходить из строя. В случае выхода из строя одного или нескольких каналов связи системный администратор может восстановить один из них.

Пара серверов называется устойчиво соединенной, если при выходе из строя одного или двух каналов связи системный администратор может восстановить один из вышедших из строя каналов, и между серверами данной пары снова можно передавать сообщения с использованием исправных каналов.

Требуется найти количество устойчиво соединенных пар серверов в сети. Пары неупорядоченные, то есть (a, b) и (b, a) — это одна и та же пара.

Формат ввода


В первой строке находятся два целых числа n и m — количество серверов и каналов связи в сети, соответственно ($2 \leq n \leq 200\,000$, $1 \leq m \leq 200\,000$).

Следующие m строк содержат по два целых числа — номера серверов, соединённых каналами связи. Гарантируется, что от любого сервера по каналам связи можно передать сообщение до любого другого, никакие два сервера не соединены более чем одним каналом связи, никакой канал связи не соединяет сервер сам с собой.


Формат вывода

Выведите одно целое число: количество устойчиво соединенных пар серверов.

Пример

Ввод 

```
8 9
1 2
2 3
1 3
1 4
4 5
5 6
6 7
7 8
5 8
```

Вывод 

16

Примечания

В примере устойчиво связаны следующие пары серверов: $(1, 2)$, $(1, 3)$, $(1, 4)$, $(2, 3)$, $(2, 4)$, $(3, 4)$, $(4, 5)$, $(4, 6)$, $(4, 7)$, $(4, 8)$, $(5, 6)$, $(5, 7)$, $(5, 8)$, $(6, 7)$, $(6, 8)$, $(7, 8)$.