



The Pycon logo features the word "Pycon" in a bold, black, sans-serif font. The text is positioned over a bright yellow parallelogram that is tilted at an angle, creating a dynamic and modern look.

Pycon



# Pycon

Добро пожаловать на Pycon — площадку для проведения олимпиад по спортивному программированию на Python.

У нас есть [задачи](#) и [контесты](#).

Разумеется, чтобы принимать участие во всех этих замечательных штуках, вам нужно [войти в систему](#), а если нет аккаунта, очевидно, [зарегистрироваться](#).

Здесь, наверное, будут новости всякие.

Наверное, здесь достаточно просто что-нибудь сломать или взломать. *Не надо, пожалуйста.*

Название читается как «Пайкон».

Оформление главной страницы — одна из самых мучительных вещей, потому что при создании обычно ничего и нет, собственно.

Но у нас есть!



## Задачи

#	Название	Решения
8	<a href="#">Самая простая задача — 2</a>	0
7	<a href="#">Самая простая задача</a>	0
6	<a href="#">Самая опасная задача</a>	1
5	<a href="#">Самая случайная задача</a>	0
4	<a href="#">Буквы C</a>	1
3	<a href="#">A / B</a>	0
2	<a href="#">A * B</a>	1
1	<a href="#">A + B</a>	2



## Задачи

#	Название	Решения
8	<a href="#">Самая простая задача — 2</a>	0
7	<a href="#">Самая простая задача</a>	0
6	<a href="#">Самая опасная задача</a>	1
5	<a href="#">Самая случайная задача</a>	0
4	<a href="#">Буквы C</a>	1
3	<a href="#">A / B</a>	1
2	<a href="#">A * B</a>	2
1	<a href="#">A + B</a>	3



# Буквы С

2 с, 64 МБ.

Учёные из Ростова-на-Дону занимаются важнейшей проблемой современности — сортировка и фильтрация строк. На этот раз им предстоит очень сложная задача — выявление особых типов букв, также известных как буквы С.

Существует цепь, состоящая из букв А и В, также называемая АБ-цепью. Учёным предстоит посчитать, сколько в этих строках встречается букв С.

## Входные данные

В первой и второй строке находятся АБ-цепи — строки, состоящие из букв А и В.

## Выходные данные

Выведите одно число — количество букв С в АБ-цепи.

## Примеры

Входные данные

```
АВВАВВВАВАВА  
ВАВАВААВАВАА
```

Выходные данные

```
0
```

Сдать

Загрузить файл

1

Отправить

[Мои посылки](#)

# Регистрация

Логин

---

Почта

---

Пароль

---

Зарегистрироваться

# Авторизация

Логин или почта

---

Пароль

---

☐ Запомнить меня

Войти



## Мои посылки

#	Дата	Название	Вердикт	Время	Память
66	12.05.2020 18:55:21	<a href="#">A * B</a>	OK	0 ms	INOP
65	12.05.2020 18:53:55	<a href="#">A * B</a>	TL #1	1005 ms	INOP
64	12.05.2020 18:53:43	<a href="#">A * B</a>	WA #1	68 ms	INOP
63	12.05.2020 18:53:32	<a href="#">A * B</a>	RE #1	68 ms	INOP
62	12.05.2020 18:53:19	<a href="#">A * B</a>	CE #1	-1 ms	INOP

# Мои посылки

#	Дата	Название	Вердикт	Время	Память
66	12.05.2020 18:55:21	<a href="#">A * B</a>	OK	0 ms	INOP
65	12.05.2020 18:53:55	<a href="#">A * B</a>	TL #1	1005 ms	INOP
64	12.05.2020 18:53:43	<a href="#">A * B</a>	WA #1	68 ms	INOP
63	12.05.2020 18:53:32	<a href="#">A * B</a>	RE #1	68 ms	INOP
62	12.05.2020 18:53:19	<a href="#">A * B</a>	CE #1	-1 ms	INOP
60	12.05.2020 18:37:39	<a href="#">Самая случайная задача</a>	WA #1	68 ms	INOP
57	12.05.2020 18:31:29	<a href="#">A + B</a>	OK	0 ms	INOP
54	12.05.2020 18:29:12	<a href="#">A + B</a>	WA #1	69 ms	INOP
53	12.05.2020 18:23:44	<a href="#">A + B</a>	CE #1	-1 ms	INOP
52	12.05.2020	<a href="#">A + B</a>	CE #1	-1 ms	INOP



# Контекст для учеников Яндекс.Лицея



Начало через 03:00:54

## [Положение](#)

# Название

A [A + B](#)

B [Самая случайная задача](#)

C [Самая простая задача — 2](#)



# Контест для учеников Яндекс.Лицея

Начало через 02:59:40

## [Положение](#)

# Название



A [A + B](#)



B [Самая случайная задача](#)

C [Самая простая задача — 2](#)

[Главная](#)[Задачи](#)[Контесты](#)Вы вошли как **Adam**[Выйти](#)

## Положение [контеста](#)



#	Имя
1	Adam
2	NeAdam
3	lol2018

=	<a href="#">A</a>	<a href="#">B</a>	<a href="#">C</a>
1	+43	-1	-1
1	+	-1	
1	+4		

**Админка**



## Задачи

#	Название	Решения	
 8	<a href="#">Самая простая задача — 2</a>	0	×
7	<a href="#">Самая простая задача</a>	0	×
 6	<a href="#">Самая опасная задача</a>	1	×
 5	<a href="#">Самая случайная задача</a>	0	×
 4	<a href="#">Буквы C</a>	1	×
3	<a href="#">A / B</a>	1	×
 2	<a href="#">A * B</a>	2	×
 1	<a href="#">A + B</a>	3	×

[Создать задачу](#)



Создать задачу

Название

Ограничения

с  МБ.

Условие

Поддерживается HTML.

Входные данные

Поддерживается HTML.

Выходные данные

Поддерживается HTML.

Создать



Создать задачу

Название

Самая простая задчка — 2

Ограничения

4

 с 

1024

 МБ.

Условие

Назовем граф с  $n$  вершинами, каждой из которых соответствует своя точка  $A_i(x_i, y_i)$  с целыми координатами, планарным деревом, если:  
Все точки  $A_1, A_2, \dots, A_n$  различны и никакие три точки не лежат на одной прямой.  
Граф является деревом, то есть в графе ровно  $n-1$  ребро и существует путь между любой парой вершин.  
Для всех пар ребер  $(i_1, j_1)$  и  $(i_2, j_2)$ , таких что  $i_1 \neq i_2, j_1 \neq j_2$ , отрезки  $A_{i_1}A_{j_1}$  и  $A_{i_2}A_{j_2}$  не пересекаются.  
Представьте планарное дерево на  $n$  вершинах. Рассмотрите выпуклую оболочку множества точек  $A_1, A_2, \dots, A_n$ . Давайте назовем это дерево паутиной если для всех  $i$  и  $j$  следующие условия верны:  
Все листья (вершины со степенью  $\leq 1$ ) лежат на выпуклой оболочке.  
Все вершины на выпуклой оболочке являются листьями.  
Точки  $A_3, A_6, A_7, A_4$  лежат на выпуклой оболочке и вершины 3,6,7,4 это все листья этого дерева.  
Обратитесь к применению для большого количества примеров.  
Давайте назовем подмножество  $S \subset \{1, 2, \dots, n\}$  вершин поддеревом, если для всех пар вершин из  $S$  существует путь между ними, содержащий только вершины из множества  $S$ . Обратите внимание, что любое поддерево планарного дерева также будет являться планарным деревом.  
Вам дано планарное дерево, состоящее из  $n$  вершин. Назовем разбиение множества вершин  $\{1, 2, \dots, n\}$  на непустые подмножества  $A_1, A_2, \dots, A_k$  (то есть  $A_i \cap A_j = \emptyset$  для любых  $i \neq j$  и  $A_1 \cup A_2 \cup \dots \cup A_k = \{1, 2, \dots, n\}$ ) хорошим разбиением, если для любых  $i$  и  $j$  подмножеств  $A_i$  и  $A_j$  выполняется следующее условие:  
Если  $A_i$  и  $A_j$  являются поддеревьями, то они не пересекаются.

Входные данные

В первой строке находится единственное целое число  $n$  ( $1 \leq n \leq 100$ ) — количество вершин в дереве.  
В следующих  $n$  строк находится по два целых числа  $x_i, y_i$  ( $-10^9 \leq x_i, y_i \leq 10^9$ ) — координаты  $i$ -й вершины, точки  $A_i$ .  
В следующей  $n-1$  строке находится по два целых числа  $x, l$  ( $1 \leq x, l \leq n$ ) — ребра  $(x, l)$  данного дерева.  
Гарантируется, что все точки различны и никакие три точки не лежат на одной прямой. Также гарантируется, что данные ребра и координаты точек описывают планарное дерево.

Выходные данные

Выведите одно целое число — количество хороших разбиений множества вершин данного планарного дерева по модулю 998244353.

Сохранить



[Посмотреть тесты](#)

[Редактировать задачу](#)

[Удалить задачу](#)



Тесты [задачи №1](#)

[Создать тест](#)

#1 ✖

Входные данные:	Выходные данные:
<input type="text" value="1 2"/>	<input type="text" value="3"/>

#2 ✖

Входные данные:	Выходные данные:
<input type="text" value="1 0"/>	<input type="text" value="1"/>

#3 ✖

Входные данные:	Выходные данные:
<input type="text" value="4 5"/>	<input type="text" value="9"/>

#6 ✖

Входные данные:	Выходные данные:
<input type="text" value="-560 10"/>	<input type="text" value="-550"/>

#7 ✖

Входные данные:	Выходные данные:
<input type="text" value="10000000000 10000000000"/>	<input type="text" value="20000000000"/>

#7 ✖

Входные данные:	Выходные данные:
<input type="text" value="0 0"/>	<input type="text" value="0"/>

#7 ✖

Входные данные:	Выходные данные:
<input type="text" value="5 -5"/>	<input type="text" value="0"/>



## Создать тест

Входные данные

Выходные данные

☐ Пример

Создать



## Контесты

#	Название	Начало	Длительность	
2	<a href="#">Контест для учеников Яндекс.Лицея</a>	12.05.2020 20:01:00	5:00:00	✗
1	<a href="#">Контест для меня</a>	12.05.2020 13:38:00	0:01:00	✗


[Создать контест](#)



## Создать контекст

Название

Начало

Длительность

☐ Скрыть список задач до начала


Создать



## Создать контекст

Название

Начало

Длительность

☐ Скрыть список задач до начала

Сохранить



[Главная](#)

[Задачи](#)

[Контесты](#)

Вы вошли как **Adam**

[Выйти](#)

## Добавить задачу

ID задачи

Добавить

# Архитектура

User

Problem

Submission

Test

Contest

UserToProblem

UserToContest

Как всё это работает

Flask, SQLAlchemy,  
multiprocessing, subprocess,  
FlaskWTF, FlaskLogin, datetime, time,  
os, random, Codemirror, functools, shutil, sys.

# Соответствие требованиям

HTML — 1038 строк.

Python — 584 исполнительных и 325 строк моделей.

CSS — 489 строк.

*(чистого и красивого кода)*

Web, ORM.



**Что будет дальше**

**Спасибо, Яндекс.**