

Git

Author: Muhammad Rengga Putra Kuncoro

Time Limit	1s
Memory Limit	256 MB



Git adalah sebuah sistem kontrol versi (version control system) yang terdistribusi, yang digunakan untuk melacak dan mengelola perubahan pada file dalam proyek perangkat lunak. Git memungkinkan pengembang untuk berkolaborasi secara efisien, menyimpan riwayat perubahan, dan mempermudah pengembangan perangkat lunak.

Ren dan Adel adalah dua mahasiswa yang mendaftarkan diri sebagai calon admin Lab Alpro. Ketika diumumkan bahwa seleksi calon admin Lab akan melibatkan proyek akhir berbasis Laravel, mereka langsung tertarik. Tantangannya bukan sekadar membangun sebuah website, tetapi juga bagaimana mereka bisa bekerja sama secara profesional menggunakan Git—alat yang banyak digunakan oleh pengembang perangkat lunak untuk mencatat dan mengelola perubahan dalam proyek. Bagi mereka, ini bukan hanya soal menulis kode, tapi juga soal belajar berkolaborasi secara efisien, terstruktur, dan bertanggung jawab.

Selama proses pengembangan, Git menjadi teman setia yang mencatat setiap langkah—mulai dari file baru yang ditambahkan, folder yang dihapus, hingga perubahan struktur yang dilakukan untuk menyempurnakan sistem. Terkadang mereka membuat kesalahan, terkadang harus memindahkan banyak bagian dari proyek, tapi semua itu tercatat dengan rapi dalam sistem Git. Perubahan-perubahan ini dikategorikan ke dalam tiga jenis:

- **Insertions** untuk penambahan folder.
- **Deletions** untuk penghapusan folder.
- **Updates** untuk setiap perpindahan lokasi folder atau mengganti nama folder.

Format Input:

Terdapat dua bagian input, yang masing-masing diakhiri dengan 0 0:

1. **Struktur direktori sebelum perubahan.**
2. **Struktur direktori setelah perubahan.**

Setiap baris mewakili hubungan antara *parent directory* dan *child directory*. Tiap folder memiliki nama yang unik, **tidak mungkin ada 2 folder dengan nama yang sama dengan parent yang berbeda**. Hanya salah satu operasi “update” yang bisa dilakukan pada tiap file, tidak mungkin keduanya (folder yang sudah diganti namanya tidak akan dipindah, begitu pula sebaliknya).

Format Output:

x updates, +y insertions, -z deletions.

Catat bahwa jika bersifat satuan (singular), maka **jangan menggunakan ‘s’** (update, insertion, deletion).

Constraints:

- $1 \leq \text{len}(\text{folder_name}) \leq 100$. Nama folder hanya berisi huruf.
- $1 \leq |E| \leq 100$, dimana E adalah relasi parent-child.

Contoh Input:

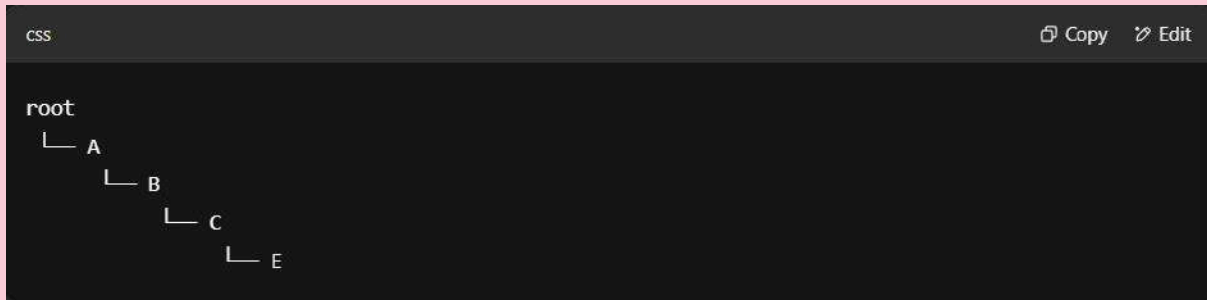
```
root A
A B
B C
C E
0 0
root B
B A
B D
D E
D F
0 0
```

Contoh Output:

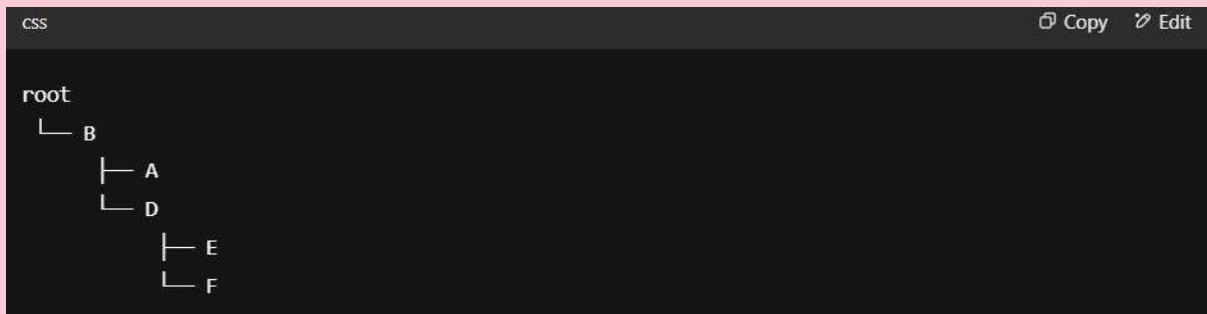
3 updates, +1 insertion, -0 deletions.

Penjelasan Output:

Struktur direktori lama:



Struktur direktori baru:



Operasi yang terjadi:

- Folder B **dipindah** ke root.
- Folder A **dipindah** ke folder B.
- Folder F **ditambahkan** ke folder C.
- Folder C **diganti namanya** ke folder D.

Git

Author: Muhammad Rengga Putra Kuncoro

Time Limit	1s
Memory Limit	256 MB



Git is a free, open-source distributed version control system used for tracking changes in computer files, primarily for source code management in software development. It allows developers to collaborate on projects, manage code versions, and revert to earlier versions if needed. Git is considered the standard version control system in the software development industry.

Ren and Adel are two students who registered as candidates for the Lab Alpro admin position. When it was announced that the selection process would involve a final project based on Laravel, they were immediately interested. The challenge was not just about building a website, but also about how they could work together professionally using Git—a tool widely used by software developers to record and manage changes within a project. For them, it wasn't just about writing code, but also about learning to collaborate efficiently, in a structured way, and with responsibility.

Throughout the development process, Git became a reliable companion, recording every step—from newly added files, deleted folders, to structural changes made to refine the system. Sometimes they made mistakes, sometimes they had to move many parts of the project, but everything was neatly recorded by the Git system. These changes are categorized into three types:

- **Insertions** for adding folders.
- **Deletions** for removing folders.
- **Updates** for any folder relocation or renaming.

Input Format:

There are two sections of input, each ending with 0 0:

1. The first section represents the directory structure **before the changes**.
2. The second section represents the directory structure **after the changes**.

Each line represents a relationship between a **parent directory** and a **child directory**. Each folder has a unique name, **there cannot be two folders with the same name under different parents**. Only one of the two update operations can be done on a single folder, never both (folder that has been renamed cannot be moved, vice versa).

Output Format:1

x updates, +y insertions, -z deletions.

Note that if the term is singular, do not use an 's' (use **update**, **insertion**, **deletion** instead).

Constraints:

- $1 \leq \text{len}(\text{folder_name}) \leq 100$. Folder name only consists of letters.
- $1 \leq |E| \leq 100$, where E is a parent-child relation.

Input Example:

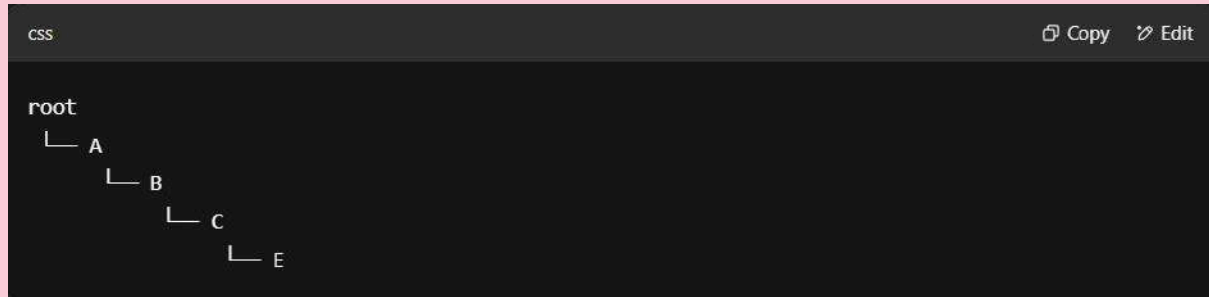
```
root A
A B
B C
C E
0 0
root B
B A
B D
D E
D F
0 0
```

Output Example:

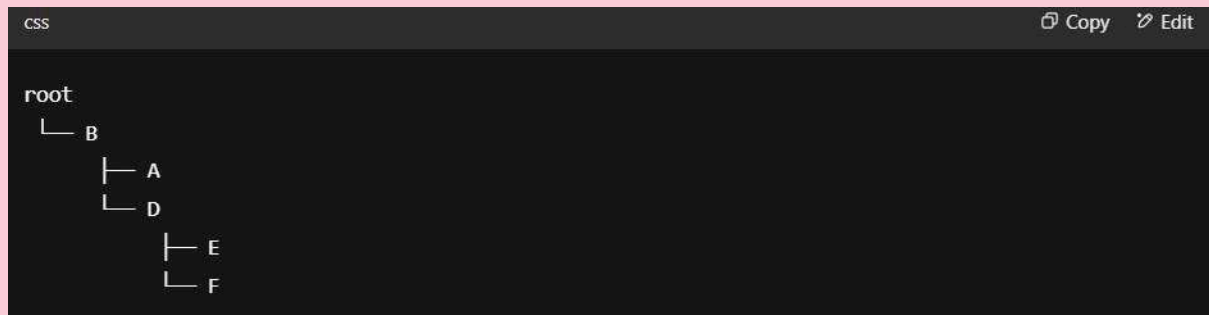
3 updates, +1 insertion, -0 deletions.

Output Explanation:

Old directory structure:



New directory structure:



Operations:

- Moved folder B to root.
- Moved folder A to B.
- Added folder F to C.
- Renamed folder C to folder D.