

1067. Структура папок

Ограничение времени: 2.0 секунды

Ограничение памяти: 64 МБ

Условие

Хакер Билл случайно потерял всю информацию с жесткого диска своего компьютера, и у него нет резервных копий его содержимого. Но он сожалеет не о потере самих файлов, а о потере очень понятной и удобной структуры папок, которую он создавал и сохранял в течение многих лет работы.

К счастью, у Билла есть несколько копий списков папок с его жесткого диска. С помощью этих списков он смог восстановить полные пути к некоторым папкам (например, WINNT\SYSTEM32\CERTSRV\CERTC0~1\X86). Он поместил их все в файл, записав каждый найденный путь в отдельную строку.

Напишите программу, которая восстановит структуру папок Билла и выведет ее в виде отформатированного дерева.

Исходные данные

Первая строка содержит целое число N – количество различных путей к папкам ($1 \leq N \leq 500$). Далее следуют N строк с путями к папкам. Каждый путь занимает одну строку и не содержит пробелов, в том числе, начальных и конечных. Длина каждого пути не превышает 80 символов. Каждый путь встречается в списке один раз и состоит из нескольких имен папок, разделенных обратной косой чертой («\»).

Имя каждой папки состоит из 1-8 заглавных букв, цифр или специальных символов из следующего списка: восклицательный знак, решетка, знак доллара, знак процента, амперсанд, апостроф, открывающаяся и закрывающаяся скобки, знак дефиса, собаки, циркумфлекс, подчеркивание, гравис, открывающаяся и закрывающаяся фигурная скобка и тильда («!#\$%&'()-@^_`{} »).

Результат

Выведите отформатированное дерево папок. Каждое имя папки должно быть выведено в отдельной строке, перед ним должно стоять несколько пробелов, указывающих на глубину этой папки в иерархии. Подпапки должны быть перечислены в лексикографическом порядке непосредственно после их родительской папки; перед их именем должно стоять на один пробел больше, чем перед именем их родительской папки. Папки верхнего уровня выводятся без пробелов и также должны быть перечислены в лексикографическом порядке.

Пример

исходные данные

```
7
WINNT\SYSTEM32\CONFIG
GAMES
WINNT\DRIVERS
HOME
WIN\SOFT
GAMES\DRIVERS
WINNT\SYSTEM32\CERTSRV\CERTC0~1\X86
```

результат

GAMES
DRIVERS
HOME
WIN
SOFT
WINNT
DRIVERS
SYSTEM32
CERTSRV
CERTCO~1
X86
CONFIG

Успешная попытка

ID	Дата	Автор	Задача	Язык	Результат проверки	№ теста	Время работы	Выделено памяти
10931620	13:35:18 9 апр 2025	Alekssei Smirnov	1067. Структура папок	Rust 1.75 x64	Accepted		0.031	4 572 КБ

Решение

на Rust 🦀

```
struct Directory {
    name: String,
    children: Vec<Directory>,
}

impl Directory {
    fn new(name: &str) -> Directory {
        return Directory { name: name.to_string(), children: vec![] };
    }

    fn add_child(&mut self, dir: Directory) {
        let _ = &self.children.push(dir);
        self.children.sort_by(|a, b| {a.name.cmp(&b.name)});
    }

    fn print_tree(&self, depth: i64, print_name: bool) {
        if print_name {
            println!("{}", " ".repeat(depth as usize), &self.name);
        }
        let _ = &self.children.iter().for_each(
            |child| {
                child.print_tree(depth+1, true);
            }
        );
    }

    fn add_by_path(&mut self, path: &mut Vec<&str>) {
        if path.len() == 0 {
            return;
        }
        let subdir_name = path.remove(0);
        for child in &mut self.children {
            if child.name == subdir_name {
                child.add_by_path(path);
                return;
            }
        }
    }
}
```

```

        }
    }
    let mut d = Directory::new(subdir_name);
    d.add_by_path(path);
    self.add_child(d);
}

}

fn main() {
    let mut root = Directory::new("root");

    let n = {
        let mut buf = String::new();
        std::io::stdin().read_line(&mut buf).expect("input");
        buf.trim().parse::<usize>().unwrap()
    };

    for _ in 0..n {
        let mut buf = String::new();
        std::io::stdin().read_line(&mut buf).expect("input");

        let mut path: Vec<&str> = buf
            .trim()
            .split("\\")
            .collect();

        root.add_by_path(&mut path);
    }

    root.print_tree(-1, false);
}

```

Про Решение

Директория:

- Имя
- Список вложенных директорий

— Основная структура в решении

Дерево из директорий.

Добавление с проверкой существования пути.