



Kurung-Mengurung

1s, 512MB

Deskripsi

Amelia Watson dan Gawr Gura sedang belajar pemrograman. Mereka menyadari bahwa pada beberapa bahasa pemrograman, tanda kurung sangat sering digunakan. Jenis kurungnya pun beragam, seperti kurung kurawal (`{}`), kurung siku (`[]`), kurung sudut (`<>`), dan kurung melengkung biasa (`()`). Ketika mereka mengetikkan source code dari program yang mereka buat, mereka sadar bahwa text editor yang mereka gunakan dapat mendeteksi tanda kurung yang tidak memiliki pasangan.

Tertarik dengan kasus tersebut, mereka berpikir untuk membuat sebuah program yang akan mendeteksi tanda kurung. Merasa tertantang, alih-alih hanya mendeteksi 1 jenis kurung, program yang mereka buat dapat mendeteksi hingga 4 jenis, yaitu kurung kurawal, kurung siku, kurung melengkung, dan kurung sudut. Mereka juga menambahkan fitur penghitung jumlah pasang kurung pada program tersebut.

Buatlah program yang dapat mendeteksi pasangan yang sesuai dari 4 jenis kurung di atas:

- kurung kurawal (`{}`),
- kurung siku (`[]`),
- kurung melengkung biasa (`()`), dan
- kurung sudut (`<>`)

Jika program mendeteksi adanya tanda kurung yang tanpa pasangan (tidak valid), maka luarannya 0. Jika semua tanda kurung telah sesuai berpasangan (valid) maka luarannya 1. Tambahkan juga fitur penghitung jumlah pasangan tiap jenis kurung.

Format Masukan

1 baris, bermacam-macam jenis kurung.

Format Keluaran

Berupa 5 baris. Baris pertama adalah n yang merupakan luaran dari deteksi tanda kurung (n = 0 jika tidak valid atau n = 1 jika valid).

Jika n = 1, baris kedua hingga kelima adalah mi yang merupakan jumlah pasang tiap kurung, dengan:

- baris ke-2: m1, jumlah pasang kurung kurawal ({})
- baris ke-3: m2, jumlah pasang kurung siku ([])
- baris ke-4: m3, jumlah pasang kurung melengkung (())
- baris ke-5: m4, jumlah pasang kurung sudut (<>)

Jika n = 0, baris kedua hingga kelima adalah mi yang semuanya bernilai nol.

Contoh Masukan (1)

```
()(){}[]<>
```

Contoh Keluaran (1)

```
1
1
1
2
1
```

Penjelasan (1)

- Masukan valid = 1
- Jumlah pasang {} = 1
- Jumlah pasang [] = 1
- Jumlah pasang () = 2
- Jumlah pasang <> = 1

Contoh Masukan (2)

```
(<{>[>])]
```

Contoh Keluaran (2)

```
0  
0  
0  
0  
0
```

Penjelasan (2)

- Masukan invalid = 0
- Karena invalid, semua pasang kurung dihitung 0

