

**Work Sheet 2**  
**Penggunaan Java Collections API**

**Struktur Data dan Algoritma**  
**IKI10400**  
**Semester Genap 2010/2011**

**Fakultas Ilmu Komputer**  
**Universitas Indonesia**

**Batas waktu pengumpulan kode sumber:**  
Senin, 7 Maret 2011 pukul 21.00 Waktu Server Aren

Kode sumber yang dinilai hanya yang dikumpulkan melalui Aren.  
Kode sumber yang dikumpulkan melalui mekanisme selain itu  
akan diabaikan dan dianggap tidak mengumpulkan.

**Peringatan: jangan mengumpulkan pekerjaan beberapa menit  
menjelang batas waktu pengumpulan karena ada kemungkinan  
pengumpulan gagal dilakukan atau koneksi internet terputus!**

Jika tidak dapat mengumpulkan tugas sebelum batas waktu  
karena suatu atau beberapa hal khusus, mahasiswa yang  
bersangkutan harus melakukan langkah-langkah dalam SOP  
Perpanjangan Batas Waktu Pengumpulan Pekerjaan.

## Tutor SDA Asyik vs. Predator

Nama berkas kode sumber : SDA11102.java

Batas waktu eksekusi program : 1 detik / kasus uji

Batas memori program : 32 MiB / kasus uji

Ada masalah besar!! Fasilkom terserang wabah Predator! Predator, merupakan singkatan Premanisme dari *Nice-tor*, adalah sebuah masalah baru yang sedang menghantui warga Fasilkom UI. *Nice-tor* si Seram selalu meminta uang setoran dari sejumlah mahasiswa setiap harinya.

Wabah Predator ini membuat Willy si Ganteng menjadi geram. Willy tidak sanggup melihat sahabat baiknya, Agung si Asyik, sedih karena tidak dapat bersepatu roda berkeliling kampus sambil mendengarkan lagu SDA ("Suara Dengarkanlah Aku" oleh ST12 *feat.* Luna Maya) kesukaannya dengan tenang karena dia takut dimintai 'setoran' oleh *Nice-tor*. Akhirnya, Willy memutuskan untuk mengeluarkan petisi yang diberi nama "Tutor SDA Asyik" (singkatan dari "Tuntutan untuk *Nice-tor* Supaya Damai, Aman, dan Anti Terusyik").

Untuk mengeluarkan petisi tersebut, Willy mengelilingi kampus dan mendata semua nama mahasiswa yang menjadi korban Predator dan jumlah 'setoran' mereka masing-masing ke *Nice-tor*. Selanjutnya, Willy membuat sebuah daftar petisi dengan ketentuan sebagai berikut.

- Petisi berupa daftar nama korban Predator yang terurut tidak menaik (*non-ascending*) berdasarkan jumlah setoran.
- Jika ada dua (atau lebih) data yang jumlah 'setoran'-nya sama, data tersebut diurutkan berdasarkan nama korban secara leksikografis menaik (*ascending*).
- Jika dalam daftar korban terdapat nama korban yang sama, maka hal itu dipastikan adalah kesalahan pemasukan data. Dengan demikian, ketika terjadi dua (atau lebih) data dengan nama yang sama (baik jumlah 'setoran' berbeda atau pun sama), petisi hanya mencantumkan data yang pertama kali muncul.
- Data yang memiliki jumlah 'setoran' 0 (nol) diabaikan karena Willy menganggap orang yang memiliki jumlah 'setoran' nol bukanlah korban Predator.

Sebagai pemrogram handal dan demi menyelesaikan masalah besar tersebut, Anda dipercayai Willy untuk membuat program yang menghasilkan petisi sesuai ketentuan di atas. Buatlah program tersebut!

### Format Masukan

Masukan, berupa hasil pendataan Willy, dibaca dari masukan standar. Masukan terdiri dari  $N$  buah baris dengan  $N \leq 100\,000$ . Masukan diproses sampai ditemukan EOF (*End-Of-File*). Setiap baris berisi sebuah datum yang memiliki format berikut.

<code>&lt;Nama&gt;, &lt;JumlahSetoran&gt;</code>
--

<**Nama**> merupakan sebuah untaian karakter yang hanya terdiri dari huruf kecil a..z. Panjang nama korban minimal 1 karakter dan maksimal 6 karakter. Sedangkan, <**JumlahSetoran**> merupakan sebuah bilangan bulat nonnegatif  $\leq 1\,000\,000$ . Karakter ',' merupakan tanda pemisah antara nama korban dan jumlah 'setoran'-nya.

### Format Keluaran

Keluaran, berupa sebuah daftar petisi sesuai ketentuan di atas, ditulis ke keluaran standar. Keluaran terdiri dari  $M$  buah baris dengan  $M \leq N$ . Setiap baris berisi sebuah informasi mengenai seorang korban dengan format berikut (untuk selanjutnya # menyatakan sebuah karakter spasi yang ditulis demikian untuk kejelasan).

```
Korban#peringkat#ke-<i>#adalah#<Nama>,#jumlah#setoran#<JumlahSetoran>
```

<**i**> merupakan sebuah bilangan bulat positif yang menyatakan urutan peringkat korban (dimulai dari 1) berdasarkan aturan di atas. Perhatikan bahwa huruf 'K' pada tulisan "Korban" merupakan huruf kapital.

### Contoh Masukan

```
nia,12000
ido,100000
willy,0
gilang,1000000
gilang,1000000
denvil,12000
irene,12000
denvil,1000
ricky,8000
```

### Contoh Keluaran

```
Korban peringkat ke-1 adalah gilang, jumlah setoran 1000000
Korban peringkat ke-2 adalah ido, jumlah setoran 100000
Korban peringkat ke-3 adalah denvil, jumlah setoran 12000
Korban peringkat ke-4 adalah irene, jumlah setoran 12000
Korban peringkat ke-5 adalah nia, jumlah setoran 12000
Korban peringkat ke-6 adalah ricky, jumlah setoran 8000
```

### Batasan

Anda **TIDAK BOLEH** membuat kode sumber yang mengimplementasikan algoritma pengurutan terhadap data masukan. Jadi, Anda harus memanfaatkan implementasi Abstract Data Type (ADT) yang sudah tersedia dalam Java Collections API untuk menghasilkan petisi tersebut. Asisten akan memeriksa kode sumber Anda apakah terdapat implementasi algoritma pengurutan yang tidak boleh digunakan.

## Petunjuk

Gunakan struktur data yang tepat untuk masalah ini! Gunakan konstruktor yang tepat dari kelas `StringTokenizer` untuk memisahkan sebuah untaian karakter berdasarkan karakter koma! Gunakan metode `compareTo()` dari kelas `String` untuk membandingkan urutan nama korban secara leksikografis, misalnya `nama1.compareTo(nama2)`.

## Kriteria Penilaian

Terdapat dua bagian penilaian, yaitu:

- penilaian penilai otomatis Aren (50%).
- penilaian *white-box review* (50%).

Komponen-komponen penilaian *white-box review* adalah sebagai berikut.

- Pemilihan struktur data yang tepat untuk menyimpan dan mengakses data (20%).
- Implementasi yang benar dari struktur data yang dipilih, termasuk implementasi penggunaan kelas `Comparator` yang benar (80%).

Jika terdapat implementasi algoritma pengurutan yang tidak boleh digunakan dalam kode sumber berdasarkan batasan di atas, total nilai WS 2 adalah 0.