

Lab 5B

Black Lizard Planet Open Recruitment

Nama berkas kode sumber : SDA<npm>L5B.java
Batas waktu eksekusi program : 3 detik / kasus uji
Batas memori program : 256 MB / kasus uji

"I don't think you would enjoy my friendship"
- Charlotte

Note : Anda diharapkan untuk mengerjakan soal tutorial 5A terlebih dahulu sebelum mengerjakan soal ini

Tahun ajaran baru telah tiba yang berarti sudah saatnya membuka penerimaan anggota baru untuk setiap ekskul di Queen's Mayfaire Academy, begitu juga dengan ekskul unik bernama "Black Lizard Planet". Black Lizard Planet merupakan ekskul baru yang dibentuk oleh lima orang siswi, yaitu Ange le Carré, Beatrice, Chise Todo, Daisy MacBean, dan Putri Charlotte. Meskipun Black Lizard Planet merupakan ekskul baru, namun ekskul ini sangatlah populer di kalangan siswa Queen's Mayfaire Academy karena kelima siswi pembentuk ekskul ini sangatlah populer, terutama Putri Charlotte yang merupakan anggota keluarga Kerajaan Albion. Karena banyaknya anggota yang mendaftar, Putri Charlotte dan keempat temannya memutuskan untuk membuat seleksi penerimaan anggota baru. Saat ini, Putri Charlotte membutuhkan bantuan Anda untuk merancang program sistem pendaftaran anggota baru. Sebagai teman baik Putri Charlotte, bantulah Putri Charlotte untuk membuat program untuk sistem penerimaan anggota baru ekskul Black Lizard Planet

Format Masukan

Program menerima input sampai end of file dengan perintah sebagai berikut :

- REGISTER;<nama>;<x>
Mendaftarkan siswa bernama <nama> dengan nilai kepintaran <x> ke dalam sistem
- RESIGN;<nama>
Menghapuskan siswa bernama <nama> dari sistem
- RETEST;<nama>;<x>
Mengubah nilai kepintaran siswa bernama <nama> menjadi <x>

Problem setter : **JCA**

- SMARTTEST
Mendapatkan nama dari (kumpulan) siswa yang memiliki nilai kepintaran tertinggi pada sistem
- RANKING
Membuat daftar peserta yang sudah terurut berdasarkan nilai kepintarannya dari yang lebih pintar ke yang kurang pintar

Format Keluaran

- REGISTER;<nama>;<nilai kepintaran>
Output : "<nama>:<nilai kepintaran> berhasil ditambahkan" (jika <nama> belum terdaftar di dalam sistem) (<nama> akan ditambahkan ke dalam sistem)
"<nama> sudah terdaftar di dalam sistem" (jika <nama> sudah terdaftar ke dalam sistem) (<nama> tidak akan ditambahkan ke dalam sistem)
- RESIGN;<nama>
Output : "<nama> mengundurkan diri" (jika <nama> terdaftar di dalam sistem) (menghapus <nama> dari sistem)
"<nama> tidak ditemukan di dalam sistem" (jika <nama> tidak terdaftar di dalam sistem)
- RETEST;<nama>;<nilai kepintaran>
Output : "<nama>:<nilai kepintaran> perubahan berhasil" (jika <nama> terdaftar di dalam sistem) (mengubah nilai kepintaran <nama> menjadi <nilai kepintaran>)
"<nama> tidak ditemukan di dalam sistem" (jika <nama> tidak terdaftar di dalam sistem)
- SMARTTEST
Output : "<siswa_1>, <siswa_2>, ..., <siswa_n> : <max>" (jika di dalam sistem terdapat siswa yang terdaftar) (<max> merupakan nilai tertinggi dari siswa yang terdaftar dalam sistem) (siswa dengan nama lebih awal secara leksikografis dicetak terlebih dahulu)
"Tidak ada siswa yang terdaftar di dalam sistem" (jika tidak ada siswa yang terdaftar di dalam sistem)
Perhatikan bahwa jumlah siswa yang mendapatkan nilai <max> bisa lebih dari satu
- RANKING
Output :
"1. <siswa_1_1>, <siswa_1_2>, <siswa_1_3>, ..., <siswa_1_k1> : <nilai_1>
2. <siswa_2_1>, <siswa_2_2>, <siswa_2_3>, ..., <siswa_2_k2> : <nilai_2>

...

n. <siswa_n_1>, <siswa_n_2>, <siswa_n_3>, ..., <siswa_n_kn> : <nilai_n> ”

(jika terdapat siswa di dalam sistem) (siswa dengan nama lebih awal secara leksikografis dicetak terlebih dahulu)

“Tidak ada siswa yang terdaftar di dalam sistem” (jika tidak ada siswa yang terdaftar di dalam sistem)

Perhatikan bahwa sebuah nilai ke-i (nilai_i) dapat dimiliki oleh lebih dari satu siswa

Contoh Masukan

```
REGISTER;Maria Florence;75
REGISTER;Sophie Mackenzie;80
REGISTER;Alicia;75
REGISTER;Shirley Collins;95
REGISTER;Elena Clay;80
REGISTER;Jahns Christian Albert;95
REGISTER;Maria Florence;99
SMARTEST
RANKING
RESIGN;Sophie Mackenzie
RETEST;Alicia;80
RANKING
```

Contoh Keluaran

Maria Florence:75 berhasil ditambahkan
Sophie Mackenzie:80 berhasil ditambahkan
Alicia:75 berhasil ditambahkan
Shirley Collins:95 berhasil ditambahkan
Elena Clay:80 berhasil ditambahkan
Jahns Christian Albert:95 berhasil ditambahkan
Maria Florence sudah terdaftar di dalam sistem
Jahns Christian Albert, Shirley Collins : 95
1. Jahns Christian Albert, Shirley Collins : 95
2. Elena Clay, Sophie Mackenzie : 80
3. Alicia, Maria Florence : 75
Sophie Mackenzie mengundurkan diri
Alicia:80 perubahan berhasil
1. Jahns Christian Albert, Shirley Collins : 95
2. Alicia, Elena Clay : 80
3. Maria Florence : 75

Hint : Jika pada soal 5A Anda membuat Binary Search Tree yang bekerja seperti TreeSet pada Java, pada soal ini, Anda bisa membuat Binary Search Tree yang bekerja seperti TreeMap pada Java

Catatan

- Anda harus membuat Binary Search Tree sendiri dan menggunakannya pada soal ini.
- Anda tidak diperbolehkan menggunakan TreeSet, TreeMap, dan Priority Queue bawaan Java
- Anda tidak diperbolehkan menggunakan Arrays.sort maupun Collections.sort pada implementasi Binary Search Tree namun Anda masih bisa menggunakannya pada pengerjaan soal ini

- Pelanggaran dari ketentuan di atas menyebabkan pekerjaan Anda pada soal ini tidak akan dinilai