

Lab 2A

Printerin.id

Nama berkas kode sumber : SDA1718L2A.java
Batas waktu eksekusi program : 3 detik / kasus uji
Batas memori program : 256 MB / kasus uji

Deskripsi

Azer, seorang mahasiswa tingkat akhir ilmu informasi di kampusnya merasa bahwa printer di kampusnya merepotkan karena harus selalu membawa *flash disk* dan membuat *flash disk* sering terkena virus. Sebagai mahasiswa ilmu informasi, maka Azer ingin mengoptimasi layanan printer di kampusnya. Ia membuat layanan printer bernama printerin.id. Namun, Azer sebagai mahasiswa tingkat akhir merasa ingin diprioritaskan. Maka ia membuat antrian printer akan diurut berdasarkan NPM, bukan berdasarkan urutan antrian yang masuk (sehingga senior dengan angka NPM lebih kecil akan diproses terlebih dahulu). Ya ini merupakan hal yang sah-sah saja karena Azer adalah pembuat layanan ini. Anda diminta membantu Azer untuk mengimplementasikan sistem antrian printer ini.

Format Masukan

Masukan dibaca dari masukan standar, yaitu melalui inputan *keyboard* pada *command prompt*. Masukan terdiri dari beberapa baris yang harus dibaca sampai *end-of-file* (<https://stackoverflow.com/questions/17678862/reading-lines-with-bufferedReader-and-checking-for-end-of-file>) yang hanya berisi salah satu dari empat perintah berikut:

1. <NPM SISWA> SUBMIT <JUMLAH HALAMAN>

- Seorang mahasiswa dengan NPM <NPM SISWA> memasukkan antrian printer untuk <JUMLAH HALAMAN> halaman.
- <NPM SISWA> akan berupa *string* alfanumerik uppercase dengan panjang persis 8 karakter dan diawali angka 1.
- <JUMLAH HALAMAN> akan berupa bilangan bulat non-negatif kurang dari 1000.
- Setiap mahasiswa tidak dapat melakukan submisi print sebelum submisi print terakhirnya selesai diproses.

2. PRINT

- Seperti yang dijelaskan sebelumnya, antrian diurutkan berdasarkan NPM.
- Printer pada layanan Printer.id dapat mencetak hingga 10 halaman sekaligus dalam satu kali print. Bila jumlah halaman dari antrian paling depan ditambah dengan antrian kedua masih tidak lebih dari 10 halaman, maka printer akan sekaligus memproses antrian kedua, dan seterusnya.
- Suatu submisi print tidak dapat dipecah (misal sumisinya adalah 5 halaman, tidak boleh dicetak 3 halaman dulu baru 2 lagi dicetak belakangan)

3. <NPM SISWA> CANCEL

- Seluruh submisi dengan npm tersebut dihapus dari antrian.

4. STATUS <NPM SISWA>

- Tentunya ketika menggunakan layanan printer, kita ingin mengetahui apakah submisi print kita sudah diproses atau belum. Printer dapat memberitahu apakah submisi sudah selesai diproses atau masih dalam antrean dengan *command* ini.

Format Keluaran

Keluaran ditulis ke keluaran standar (*command prompt*). Untuk masing-masing perintah keluarannya adalah:

1. <NPM SISWA> SUBMIT <JUMLAH HALAMAN> HALAMAN

- Bila <JUMLAH HALAMAN> melebihi 10, cetak sebaris string:
"Jumlah halaman submisi <NPM SISWA> terlalu banyak"
- Bila <NPM SISWA> sudah ada dalam antrean, cetak sebaris string:
"Harap tunggu hingga submisi sebelumnya selesai diproses"
- Bila masukan valid, cetak sebaris string
"Submisi <NPM SISWA> telah diterima"

2. PRINT

- Bila antrean kosong, cetak sebaris string: "Antrean kosong"
- Cetak semua npm yang diproses dalam sekali print tersebut. Bila dalam satu kali print ini printer mencetak 2 submisi sekaligus, cetak 2 baris:
"Submisi <NPM SISWA PERTAMA> telah dicetak sebanyak <JUMLAH HALAMAN> halaman"
"Submisi <NPM SISWA KEDUA> telah dicetak sebanyak <JUMLAH HALAMAN> halaman"
- Bila printer melakukan 4 submisi sekaligus, cetak 4 baris, dan seterusnya.

3. <NPM SISWA> CANCEL

- Bila <NPM SISWA> tidak ada dalam atrean, keluarkan sebaris string:
"<NPM SISWA> tidak ada dalam antrean"
- Bila <NPM SISWA> ada dalam atrean, keluarkan sebaris string:
"Submisi <NPM SISWA> dibatalkan"

4. STATUS <NPM SISWA>

- Bila <NPM SISWA> belum pernah melakukan submisi print, keluarkan sebaris string:
"<NPM SISWA> tidak ada dalam sistem"
- Bila submisi <NPM SISWA> sudah ada yang selesai diproses (meskipun NPM ini bisa saja juga ada dalam antrean), keluarkan sebaris string:
"Submisi <NPM SISWA> sudah diproses"
- Bila <NPM SISWA> masih dalam antrean, keluarkan sebaris string:
"Submisi <NPM SISWA> masih dalam antrean"

PERHATIAN: Jangan melakukan copy-paste contoh input dan output dari pdf karena character encodingnya bisa berbeda. Gunakan file **input_output_lab2a.txt**

Contoh Masukan

```
14065435 SUBMIT 5 HALAMAN
13065435 SUBMIT 4 HALAMAN
14065436 SUBMIT 3 HALAMAN
13065435 SUBMIT 2 HALAMAN
PRINT
STATUS 14065435
STATUS 14065436
12065435 SUBMIT 10 HALAMAN
11065435 SUBMIT 12 HALAMAN
PRINT
14095435 SUBMIT 3 HALAMAN
14065436 CANCEL
STATUS 16095435
```

Contoh Keluaran

```
Submisi 14065435 telah diterima
Submisi 13065435 telah diterima
Submisi 14065436 telah diterima
Harap tunggu hingga submisi sebelumnya selesai diproses
Submisi 13065435 telah dicetak sebanyak 4 halaman
Submisi 14065435 telah dicetak sebanyak 5 halaman
Submisi 14065435 sudah diproses
Submisi 14065436 masih dalam antrean
Submisi 12065435 telah diterima
Jumlah halaman submisi 11065435 terlalu banyak
Submisi 12065435 telah dicetak sebanyak 10 halaman
Submisi 14095435 telah diterima
Submisi 14065436 dibatalkan
16095435 tidak ada dalam sistem
```

Lab 2B

Kolektor Mobil

Nama berkas kode sumber : SDA1718L2A.java
Batas waktu eksekusi program : 3 detik / kasus uji
Batas memori program : 256 MB / kasus uji

Deskripsi

Pak Mihaf adalah seorang kolektor mobil. Ia memiliki begitu banyak mobil di parkirannya, namun garasi Pak Mihaf sangatlah sempit dan pintu garasi dari Pak Mihaf hanya ada 2, pintu barat dan pintu timur. Sehingga, bila ia ingin mengeluarkan mobil ia perlu mengeluarkan semua mobil lain yang berada diantara mobil tersebut dan pintu garasi tersebut.

Format Masukan

Masukan dibaca dari masukan standar, yaitu melalui inputan *keyboard* pada *command prompt*. Masukan terdiri dari beberapa baris yang harus dibaca sampai *end-of-file* (EOF) yang hanya berisi salah satu dari empat perintah berikut:

1. MASUK <JENIS MOBIL> <BARAT/TIMUR>

- Sebuah mobil dengan tipe <JENIS MOBIL> masuk melalui pintu barat atau pintu timur
- <JENIS MOBIL> akan berupa string alfanumerik (tidak ada spasi) *uppercase* dengan panjang kurang dari 20 karakter.
- <JENIS MOBIL> dalam garasi dipastikan tidak ada yang sama.

2. KELUARKAN <JENIS MOBIL>

- Keluarkan mobil dengan tipe <JENIS MOBIL> dari garasi dan harus melalui pintu terdekat, baik itu pintu barat mau pun timur.
- Ketika ingin mengeluarkan suatu mobil, maka seluruh mobil lain yang berada diantara mobil tersebut dengan pintu terdekat harus dikeluarkan terlebih dahulu, lalu dimasukkan kembali. Jika jarak mobil ke pintu barat dan timur sama, mobil akan keluar melalui pintu **barat**.
- Bila mobil tidak bisa keluar melalui pintu terdekat (karena ada mobil lain yang mogok), maka keluarkan dari pintu yang lainnya

3. MOGOK <JENIS MOBIL>

- Mobil dengan tipe <JENIS MOBIL> mogok, sehingga tidak bisa digerakkan sama sekali.

4. SERVIS <JENIS MOBIL>

- Mobil dengan tipe <JENIS MOBIL> sudah diservis, sehingga bisa digerakkan kembali.

Format Keluaran

Untuk perintah:

1. MASUK <JENIS MOBIL> <BARAT/TIMUR>

- o Keluarkan sebaris string:
“<JENIS MOBIL> masuk melalui pintu <BARAT/TIMUR>”

2. KELUARKAN <JENIS_MOBIL>

- o Jika mobil tersebut tidak ada di garasi, keluarkan sebaris string:
“<JENIS MOBIL> tidak ada di garasi”
- o Jika mobil tersebut mogok, keluarkan sebaris string:
“Mobil <JENIS MOBIL> sedang mogok”
- o Jika mobil lain yang berada diantara mobil tersebut dengan pintu barat dan pintu timur ada yang mogok (sehingga ia terjepit oleh setidaknya 2 mobil mogok), keluarkan sebaris string:
“<MOBIL YANG INGIN KELUAR> tidak bisa keluar, mobil <MOBIL MOGOK TERDEKAT DI SISI BARAT> dan <MOBIL MOGOK TERDEKAT DI SISI TIMUR> sedang mogok” (lihat penjelasan contoh di bawah).
- o Jika mobil tersebut berhasil keluar, keluarkan sebaris string:
“<JENIS MOBIL> keluar melalui pintu <BARAT/TIMUR>”

3. MOGOK <JENIS MOBIL>

- o Tidak perlu mengeluarkan keluaran apapun.

4. SERVIS <JENIS MOBIL>

- o Tidak perlu mengeluarkan keluaran apapun

PERHATIAN: Jangan melakukan copy-paste contoh input dan output dari pdf karena character encodingnya bisa berbeda. Gunakan file **input_output_lab2b.txt**

Contoh Masukan

```
MASUK LAMBORGHINI BARAT
MASUK FERRARI BARAT
MASUK ROYALROYCE TIMUR
MASUK TOYOTA TIMUR
MASUK HONDA BARAT
KELUARKAN ROYALROYCE
MOGOK TOYOTA
KELUARKAN LAMBORGHINI
MASUK CHEVROLET BARAT
SERVIS TOYOTA
KELUARKAN FERRARI
MOGOK TOYOTA
MOGOK CHEVROLET
MASUK DAIHATSU BARAT
MOGOK DAIHATSU
KELUARKAN HONDA
```

Contoh Keluaran

```
LAMBORGHINI masuk melalui pintu BARAT
FERRARI masuk melalui pintu BARAT
ROYALROYCE masuk melalui pintu TIMUR
TOYOTA masuk melalui pintu TIMUR
HONDA masuk melalui pintu BARAT
ROYALROYCE keluar melalui pintu TIMUR
LAMBORGHINI keluar melalui pintu BARAT
CHEVROLET masuk melalui pintu BARAT
FERRARI keluar melalui pintu TIMUR
DAIHATSU masuk melalui pintu BARAT
HONDA tidak bisa keluar, mobil CHEVROLET dan TOYOTA sedang mogok
```

Penjelasan Contoh

- Di sisi barat mobil ROYALROYCE terdapat mobil LAMBORGHINI, FERRARI, dan HONDA sedangkan di sisi timur mobil ROYALROYCE hanya terdapat mobil TOYOTA, maka ROYALROYCE keluar melalui pintu TIMUR.
- Mobil LAMBORGHINI lebih dekat untuk keluar melalui pintu TIMUR, namun karena mobil TOYOTA sedang mogok, LAMBORGHINI keluar melalui pintu BARAT.
- Ketika mobil HONDA ingin keluar, di sisi barat terdapat mobil DAHATSU dan CHEVROLET yang mogok dandi sisi timur terdapat mobil TOYOTA yang juga mogok. Maka dari itu mobil HONDA tidak dapat keluar garasi, dan karena mobil HONDA lebih dekat dengan mobil CHEVROLET dibandingkan mobil HONDA, maka yang dicetak hanyalah pemberitahuan mogok untuk CHEVROLET dan TOYOTA.