



Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)
Tahun / Semester : 2020-2021 / Genap
Modul : 7 - *Stacks and Queues*
Hari, Tanggal Praktikum : Rabu, 7 April 2021

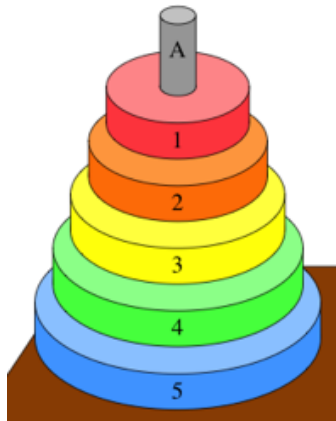
Naskah Soal Praktikum

Pembuat Naskah: David Fauzi, Alexandra Handayani

Ketentuan:

1. Kerjakanlah satu dari dua soal berikut pada *template repository* yang anda peroleh ketika mengambil *assignment* di GitHub Classroom praktikum!
2. *Commit* yang dilakukan setelah sesi praktikum berakhir tidak akan dipertimbangkan dalam penilaian.
3. *Header* setiap *file* harus mengikuti format yang telah disediakan pada *file template repository*. *Header* yang tidak mengikuti format tersebut tidak akan dinilai.
4. Buku catatan laboratorium yang berisi *flowchart*, *data flow diagram*, dan analisis kompleksitas waktu dan ruang Big O dari solusi yang anda buat dikumpulkan ke tugas.stei.itb.ac.id paling lambat 24 jam setelah sesi praktikum. Tulis pula alasan anda memilih mengerjakan soal yang anda kerjakan dan mengapa bukan soal yang lainnya!
5. Solusi soal pertama harus dapat dikompilasi dengan perintah `make soal-01` dan menghasilkan *file executable* dengan nama `soal-01`. Demikian pula, soal kedua harus dapat dikompilasi dengan perintah `make soal-02` dan menghasilkan *file executable* dengan nama `soal-02`.
6. Bila diperlukan, sesuaikanlah isi *Makefile* yang tersedia pada *template repository* untuk memenuhi syarat kompilasi dan *file* keluaran di atas!

Soal 1



Kak Mikiya merupakan seorang lulusan *fresh graduate* teknik sipil yang diberikan proyek untuk membangun suatu *tower* yang terdiri dari N buah piringan dengan ukuran masing-masing piringan berbeda di rentang 1 sampai N . setiap harinya, Ia diberikan 1 buah piringan dengan ukuran yang acak (tentunya di rentang 1- N). Kak Mikiya diberikan waktu N hari untuk menyelesaikan *tower* tersebut. Tugas utama dari kak Mikiya adalah memberikan arahan ke kuli bangunan mengenai piringan mana saja yang harus dipasang ke *tower* untuk setiap harinya. Berikut adalah *rules* dari penempatan piringan :

- Setiap harinya, anda diberikan 1 piringan dengan ukuran berbeda/unik di rentang 1- N
- Setiap piringan didalam *tower* berukuran *distinct* (berbeda) dan *tower* akhir harus terdiri dari susunan piringan yang mengurut dari ukuran N (paling bawah) hingga ukuran 1 (paling atas).

***lihat contoh gambar**

- Piringan dengan ukuran yang lebih besar harus ditaruh di bagian bawah *tower*, sedangkan piringan dengan ukuran lebih kecil harus ditaruh di bagian atas *tower*
- Anda tidak boleh menaruh piringan baru ke atas *tower* sampai seluruh piringan dengan ukuran yang lebih besar dari piringan baru tersebut telah ditaruh di *tower*

Program akan menerima input berupa N yaitu banyaknya piringan yang akan diberikan didalam N buah hari. Program kemudian akan menerima daftar ukuran piringan yang diberikan untuk setiap harinya. Program akan memberikan output berupa piringan dengan ukuran apa yang harus dipasang untuk setiap harinya dari hari pertama hingga hari ke- N . Jika pada hari tersebut tidak ada piringan yang bisa ditaruh, maka berikan output “-“ pada hari tersebut.

Keterangan :

- Penyelesaian masalah diharuskan untuk menggunakan tipe data *queue* pada program. Input awal dapat dimasukan dahulu ke dalam *array* dan baru diproses menggunakan *queue*.

- **Perhatikan format output**, tidak boleh memberikan karakter spasi di akhir baris karena akan dianggap salah ketika diperiksa oleh autograder.

Contoh eksekusi program (garis bawah menunjukkan input)

#1

Jumlah hari: 5
Ukuran piringan yang diterima di day 1: 4
Ukuran piringan yang diterima di day 2: 5
Ukuran piringan yang diterima di day 3: 1
Ukuran piringan yang diterima di day 4: 2
Ukuran piringan yang diterima di day 5: 3
Tahapan Pembangunan Tower:
Day 1: -
Day 2: 5 4
Day 3: -
Day 4: -
Day 5: 3 2 1

#2

Jumlah hari: 4
Ukuran piringan yang diterima di day 1: 3
Ukuran piringan yang diterima di day 2: 2
Ukuran piringan yang diterima di day 3: 1
Ukuran piringan yang diterima di day 4: 4
Tahapan Pembangunan Tower:
Day 1: -
Day 2: -
Day 3: -
Day 4: 4 3 2 1

#3

Jumlah hari: 8
Ukuran piringan yang diterima di day 1: 1
Ukuran piringan yang diterima di day 2: 5
Ukuran piringan yang diterima di day 3: 6
Ukuran piringan yang diterima di day 4: 3
Ukuran piringan yang diterima di day 5: 2
Ukuran piringan yang diterima di day 6: 8
Ukuran piringan yang diterima di day 7: 7
Ukuran piringan yang diterima di day 8: 4
Tahapan Pembangunan Tower:
Day 1: -
Day 2: -
Day 3: -
Day 4: -
Day 5: -
Day 6: 8

Day 7: 7 6 5 Day 8: 4 3 2 1

Soal 2

Seorang mahasiswa ingin mendesain suatu *browser* sederhana yang ia namakan “Mozarella Firefly”. Fitur *browser* secara umum terdiri atas fitur *back* dan *forward*. Fitur *back* akan mengembalikan ke laman yang sebelumnya dikunjungi, sedangkan fitur *forward* akan menuju laman lama yang ditinggalkan ketika fitur *back* digunakan. Untuk *interface*-nya sendiri, digunakan dengan memasukkan *character* sebagai instruksi ketika mengaktifkan fitur-fitur. Terdapat empat buah instruksi yang digunakan dalam *browser* ini :

1. ‘I’ : perintah *input* untuk memasukkan URL yang hendak dituju
2. ‘U’ : perintah *undo* untuk memasukkan fitur *back*
3. ‘R’ : perintah *redo* untuk memasukkan fitur *forward*
4. ‘E’ : perintah *exit* untuk keluar dari *browser*

Ketentuan tambahan untuk desain *browser* tersebut adalah sebagai berikut :

1. Asumsikan *character* instruksi yang dimasukkan selalu benar
2. Harus menggunakan konsep *stack*
3. URL yang dimasukkan adalah berupa *string*
4. Ketika belum ada laman yang dituju, maka fitur *back* dan *forward* tidak dapat dijalankan
5. Fitur *forward* dapat dijalankan jika fitur *back* telah dijalankan sebelumnya
6. Jika fitur *back* dipanggil dan kemudian pengguna memasukkan URL baru, maka fitur *forward* tidak akan dapat dijalankan lagi hingga fitur *back* kembali dipanggil

Bantulah mahasiswa tersebut mendesain *browser*-nya dengan menggunakan bahasa C!

Contoh eksekusi program (garis bawah menandakan input)

```
#1
-----Mozarella Firefly-----
Instruction : U
Can't Go Back!
Nothing to found ..

Instruction : R
Can't Go Forward!
Nothing to found ..
```

Instruction : I

Insert URL : www.ldte.stei.itb.ac.id

Current Site : www.ldte.stei.itb.ac.id

Instruction : E

... Exit

#2

-----Mozarella Firefly-----

Instruction : I

Insert URL : www.ldte.stei.itb.ac.id

Current Site : www.ldte.stei.itb.ac.id

Instruction : I

Insert URL : www.praktikum.stei.itb.ac.id

Current Site : www.praktikum.stei.itb.ac.id

Instruction : U

Current Site : www.ldte.stei.itb.ac.id

Instruction : E

... Exit

#3

-----Mozarella Firefly-----

Instruction : I

Insert URL : www.ldte.stei.itb.ac.id

Current Site : www.ldte.stei.itb.ac.id

Instruction : I

Insert URL : www.praktikum.stei.itb.ac.id

Current Site : www.praktikum.stei.itb.ac.id

Instruction : U

Current Site : www.ldte.stei.itb.ac.id

Instruction : R

Current Site : www.praktikum.stei.itb.ac.id

Instruction : E
... Exit

#4

-----Mozarella Firefly-----

Instruction : I

Insert URL : www.ldte.stei.itb.ac.id

Current Site : www.ldte.stei.itb.ac.id

Instruction : I

Insert URL : www.praktikum.stei.itb.ac.id

Current Site : www.praktikum.stei.itb.ac.id

Instruction : U

Current Site : www.ldte.stei.itb.ac.id

Instruction : I

Insert URL : www.wikipedia.com

Current Site : www.wikipedia.com

Instruction : R

Can't Go Forward!

Current Site : www.wikipedia.com

Instruction : E

... Exit