



Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)

Tahun / Semester : 2021-2022 / Genap

Modul : 7 - Stacks and Queues

Hari, Tanggal Praktikum : Selasa, 5 April 2022

Naskah Soal Praktikum

Pembuat Naskah: Harry Purba, Rizki Habibi Alamsyah

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 alasan pemilihan soal, *flowchart*, dan *data flow diagram* dari solusi yang anda buat dikumpulkan ke tugas.stei.itb.ac.id paling lambat pukul 11.00 WIB satu hari kerja setelah sesi praktikum.
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

Buatlah program yang dapat menyimpan *stack* buku yang akan di-*checkout* oleh *user*. *Stack* buku awal adalah *stack* yang kosong, dan memunculkan pesan bahwa daftar *checkout* buku masih kosong.

Program akan memiliki beberapa *command* berupa character untuk menambahkan buku, menghapus buku, menampilkan daftar *checkout*, dan *checkout* buku:

- “A” / *add* digunakan bila *user* ingin menambahkan buku dan penulis buku dengan format “NamaBuku, PenulisBuku”. Asumsi input penambahan buku selalu benar.
- “D” / *delete* digunakan bila *user* ingin menghapus buku paling bawah pada daftar *checkout* (buku paling bawah adalah buku yang paling terakhir ditambahkan). Bila *stack* masih kosong, program akan memberitahu bahwa daftar *checkout* masih kosong.
- “S” / *show* digunakan untuk menampilkan *stack* atau daftar *checkout* yang ada
- “C” / *Checkout* digunakan bila *user* ingin *checkout* buku. Program akan menampilkan daftar *checkout* dan keluar dari program. Bila daftar *checkout* masih kosong, program tidak dapat melakukan *checkout*.
- “E” / *Exit* digunakan bila *user* ingin keluar dari program tanpa *checkout* buku.
- Selain dari *command* itu, program akan memberitahu bahwa *command* yang dikirim salah.

Program akan terus *loop* sampai *user* melakukan *Exit* atau *Checkout* dengan daftar *checkout* yang terisi.

Hint : Gunakan `fgets` dan `strtok` untuk *parsing input* nama buku dan penulis buku.

Contoh Eksekusi Program (garis bawah menandakan input)

```
#1
Program Checkout Buku
Masukkan Perintah: A

Masukkan Nama Buku dan Penulis Buku: Harry Potter, J. K. Rowling

Masukkan Perintah: A

Masukkan Nama Buku dan Penulis Buku: Percy Jackson, Rick Riordan

Masukkan Perintah: A

Masukkan Nama Buku dan Penulis Buku: Hunger Games, Suzanne Collins

Masukkan Perintah: S
```

Checkout Buku:
Hunger Games by Suzanne Collins
Percy Jackson by Rick Riordan
Harry Potter by J. K. Rowling

Masukkan Perintah: D

Masukkan Perintah: C

Berikut adalah daftar buku yang akan dipinjam!

Checkout Buku:
Percy Jackson by Rick Riordan
Harry Potter by J. K. Rowling

Terimakasih sudah menggunakan layanan Checkout Buku!

#2

Program Checkout Buku
Masukkan Perintah: C

Checkout Buku:
Daftar Checkout Masih Kosong!
Masukkan Perintah: U

Input Perintah Salah! Silahkan Coba Lagi!
Masukkan Perintah: A

Masukkan Nama Buku dan Penulis Buku: How to PPMC,Dismas Widyanto

Masukkan Perintah: A

Masukkan Nama Buku dan Penulis Buku: Asisten Terbaik,Kevin Naoko

Masukkan Perintah: C

Berikut adalah daftar buku yang akan dipinjam!

Checkout Buku:
Asisten Terbaik by Kevin Naoko
How to PPMC by Dismas Widyanto

Terimakasih sudah menggunakan layanan Checkout Buku!

#3

Program Checkout Buku

Masukkan Perintah: D

Checkout Buku:

Daftar Checkout Masih Kosong!

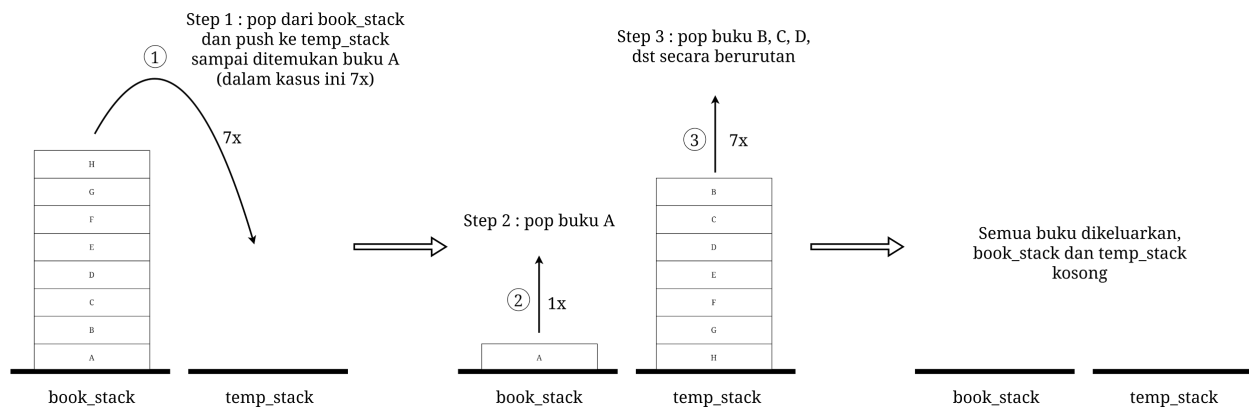
Masukkan Perintah: E

Terimakasih sudah menggunakan layanan Checkout Buku!

Soal 2

Anda diberikan dua tempat tumpukan buku yang dimodelkan menggunakan *stack*, yaitu *book_stack* dan *temp_stack*. Pada kondisi awal, *book_stack* berisi 8 tumpukan buku dengan judul A, B, C, D, E, F, G, dan H yang tersusun secara acak. Urutan tumpukan buku dari atas ke bawah diperoleh dari *input user*. Anda diminta untuk mengeluarkan buku secara berurutan dari A sampai H. Anda dapat melakukan dua operasi pada tumpukan, yaitu mengangkat buku (*pop* buku dari *stack*) atau meletakkan buku (*push* buku ke dalam *stack*). Kedua operasi *pop* dan *push* dapat dilakukan baik pada *book_stack* maupun *temp_stack*. Untuk *temp_stack*, tumpukan ini pada awalnya kosong, fungsinya adalah sebagai tempat perantara meletakkan buku saat melakukan pencarian buku di *book_stack* karena anda tidak boleh melakukan operasi *pop* secara berurutan atau dengan kata lain anda hanya dapat memegang satu buku dalam satu waktu.

Misalkan diberikan input urutan *book_stack* dari atas ke bawah: HGFEDCBA maka ilustrasi langkah yang dilakukan dapat dilihat pada gambar berikut.



Tugas anda adalah menghitung berapa jumlah tumpukan buku tertinggi pada *temp_stack* selama proses tersebut berlangsung sampai semua buku dikeluarkan, misal pada kasus di atas, tumpukan *temp_stack* tertinggi adalah sebanyak 7 buku, sehingga akan menghasilkan *output* 7. Buatlah program yang dapat menghitungnya dengan urutan *book_stack* dari *input user*.

Catatan :

- *Input* berupa *string* sepanjang 8 karakter huruf A-H yang menyatakan urutan *book_stack* dari atas ke bawah, *input* diasumsikan *valid*.
- Pemodelan *book_stack* dan *temp_stack* harus menggunakan struktur data *stack*, tidak boleh menggunakan *array*. Operasi *stack* yang dilakukan adalah *push* dan *pop* (implementasikan menggunakan fungsi). Implementasikan juga fungsi untuk menghitung tinggi *temp_stack*.

- Pencarian buku dimulai dari `book_stack` menggunakan operasi `pop`, jika ditemukan buku yang bukan sesuai urutannya, maka `push` buku tersebut ke `temp_stack`. Jika buku pada `book_stack` habis, selanjutnya lakukan pencarian pada `temp_stack`, pada kondisi ini, jika buku yang ditemukan bukan buku yang sesuai, maka boleh di `push` ke `book_stack`, sehingga pada iterasi selanjutnya pencarian buku kembali dilakukan dari `book_stack`. Dengan kata lain, prioritas pencarian buku selalu dimulai dari `book_stack` lalu `temp_stack`.
- *Output* berupa *integer* yang menunjukkan tinggi atau jumlah tumpukan buku maksimum pada `temp_stack`

Contoh Eksekusi Program (garis bawah menandakan input)

#1

Masukkan urutan `book_stack`: HGFEDCBA

Tinggi maksimum `temp_stack`: 7

#2

Masukkan urutan `book_stack`: ABCDEFGH

Tinggi maksimum `temp_stack`: 0

#3

Masukkan urutan `book_stack`: EFGHABCD

Tinggi maksimum `temp_stack`: 4