



Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)
Tahun / Semester : 2022-2023 / Genap
Modul : 7 - Stacks and Queues
Hari, Tanggal Praktikum : Jumat, 31 Maret 2023

Naskah Soal Praktikum

Pembuat Naskah: Emmanuella Rumanti, Reynaldo Averill

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* dan *data flow diagram* dari solusi yang anda buat dikumpulkan ke praktikum.stei.itb.ac.id paling lambat pukul 11.00 WIB dua hari kerja setelah sesi praktikum. Tulis pula alasan anda memilih mengerjakan soal yang anda kerjakan dan penjelasan/rancangan dari algoritma yang Anda gunakan!
5. Solusi soal pertama dan kedua harus dapat dikompilasi dengan perintah `make main` dan menghasilkan *file executable* dengan nama `main`.
6. Bila diperlukan, sesuaikanlah isi *Makefile* yang tersedia pada *template repository* untuk memenuhi syarat kompilasi dan *file* keluaran di atas!

Soal 1

Rahm adalah seorang mahasiswa nerd yang berencana untuk menjadi ketua UNI alias Unit Nerd ITB. Untuk membuktikan ke-*nerdy*-annya, Rahm merancang bahasa pemrograman bernama Ebstfelfkkeb yang *interpreter*-nya ditulis menggunakan bahasa C agar cepat.



Gambar 1-1 Ebstfelfkkeb

Salah satu hal yang Rahm mesti implementasikan adalah mengecek buka-tutup kurung kode. Sebuah program bahasa Ebstfelfkkeb hanya dapat berjalan apabila semua dari ketiga jenis kurung, yaitu `()` `[]` dan `{}` dibuka dan ditutup sesuai urutannya.

<code>()[]() -> OKE</code>	<code>{{()}} -> OKE</code>	<code>({}) -> TIDAK OKE</code>	<code>{{{{{}} -> TIDAK OKE</code>
-------------------------------	-------------------------------	-----------------------------------	--------------------------------------

Dari studi literatur, Rahm menemukan bahwa salah satu cara untuk memverifikasi kurung pada kode adalah dengan menggunakan stack.

- Verifikator membaca string karakter per karakter.
- Jika menemukan kurung buka, kurung tersebut di-*push* ke dalam stack.
- Jika menemukan kurung tutup, dicek apakah kurung tersebut “cocok” dengan kurung yang terakhir ditambahkan ke stack. Jika cocok, dilakukan *pop* pada stack.
- Kurung pada string sudah sesuai jika di akhir komputasi, stack tersebut kosong.

Bantu Rahm menjadi ketua UNI dengan mengimplementasikan verifikasi kurung untuk interpreter Ebstfelfkkeb menggunakan metode stack! Silakan menggunakan dan memodifikasi fungsi yang diberikan pada template.

Contoh Eksekusi Program (garis bawah menandakan input):

#1

Masukkan string: {[[[(())]]}{[()]}]
Semua kurung sesuai

#2

Masukkan string: ()()()()()() (
Ada kurung tidak sesuai

Soal 2



Sebagai salah satu siswa di sekolah sihir *Hokwartz*, Feijun harus menghadapi ujian pada kelas Transfigurasi (Ilmu tentang perubahan objek atau manusia menggunakan sihir). Pada ujian ini, ia dihadapkan dengan sebuah objek antik bernama *Tablet of Founder* yang tersusun dari 49 buah permata, tersusun dalam persegi dengan susunan 7×7 . *Tablet of Founder* telah berusia ratusan tahun, sehingga sudah tercemar oleh berbagai macam sihir yang mengakibatkan berubahnya bentuk dari *Tablet of Founder*. Target dari ujian kelas transfigurasi adalah mengubah bentuk dari *Tablet of Founder* dengan mantra yang dipelajari dari kelas transfigurasi. Kedua materi yang dapat digunakan beserta fungsinya adalah:

1. *Undounus Incantatem* : Menetralkan satu matara yang diterapkan terakhir kali pada *Tablet of Founder* (Anggap seperti fungsi *Undo* mantra).
2. *Redolus Revelio*: Memunculkan kembali satu mantra yang telah dinetralkan dengan *Undounus Incantatem* (Anggap seperti fungsi *Redo* mantra).

Dengan latar belakangnya sebagai *muggleborn* (istilah untuk penyihir yang lahir dari keturunan bukan penyihir), Feijun ingin menggunakan bahasa C untuk menyimulasikan kondisi tablet setelah diberikan berbagai mantra transfigurasi. Bantulah Feijun untuk membuat program dalam bahasa C yang dapat menyimulasikan ujian transfigurasi Feijun! **Program harus menggunakan konsep *Stack* atau *Queue*.**

Tablet of Founder dimodelkan dengan matriks berukuran 7×7 dengan tipe data *char*. *Tablet of Founder* selalu memiliki kondisi awal yang sama. Perubahan *Tablet of Founder* disimpan pada sebuah file dengan format .txt yang berisi jumlah perubahan yang terjadi, indeks dari *Tablet of Founder* yang mengalami perubahan (indeks baris dan kolom dimulai dari 0, diakhiri dengan baris dan kolom ke 6). Kemudian, mantra *Undounus Incantatem* (dilambangkan sebagai u pada program) dan *Redolus Revelio* (dilambangkan sebagai r pada program) digunakan untuk melakukan *undo* atau *redo* terhadap kondisi *Tablet of Founder*. Berikut adalah contoh alur perubahan kondisi *Tablet of Founder* (ToF) seiring keberjalanan program.

Kondisi Awal <i>ToF</i>						
O	*	*	-	*	*	O
*	X	*	-	*	X	*
*	*	X	-	X	*	*
-	-	-	X	-	-	-
*	*	X	-	X	*	*
*	X	*	-	*	X	*
O	*	*	-	*	*	O

Kondisi <i>ToF</i> hingga perubahan ke 5						
X	*	*	-	*	*	-
*	X	*	-	*	X	*
*	*	X	-	X	*	*
-	-	-	O	-	-	-
*	*	X	-	X	*	*
*	X	*	-	*	X	*
X	*	*	-	*	*	X

Kondisi <i>ToF</i> hingga perubahan ke 7 (<i>ToF</i> sebelum U dan R)						
X	*	*	-	*	*	X
*	X	*	-	*	X	*
*	*	X	-	X	*	*
-	-	-	*	-	-	-
*	*	X	-	X	*	*
*	X	*	-	*	X	*
X	*	*	-	*	*	X

Kondisi <i>ToF</i> setelah u 3 kali (Kembali ke setelah perubahan ke 4)						
X	*	*	-	*	*	-
*	X	*	-	*	X	*
*	*	X	-	X	*	*
-	-	-	O	-	-	-
*	*	X	-	X	*	*
*	X	*	-	*	X	*
X	*	*	-	*	*	O

<i>ToF</i> saat u 3 kali lanjut r 1 kali (Maju ke setelah perubahan ke 5)						
X	*	*	-	*	*	-
*	X	*	-	*	X	*
*	*	X	-	X	*	*
-	-	-	O	-	-	-
*	*	X	-	X	*	*
*	X	*	-	*	X	*
X	*	*	-	*	*	X

Kondisi akhir <i>ToF</i>						
X	*	*	-	*	*	-
*	X	*	-	*	X	*
*	*	X	-	X	*	*
-	-	-	O	-	-	-
*	*	X	-	X	*	*
*	X	*	-	*	X	*
X	*	*	-	*	*	X

Beberapa hal yang harus diketahui terkait program yang dibuat adalah sebagai berikut:

- Pembacaan dari file eksternal telah dibuat pada template.** Seluruh fungsi yang diperlukan juga sudah dideklarasikan pada template, sehingga hanya perlu membuat implementasinya serta pemanggilannya pada program utama. Tetapi, silahkan tambahkan fungsi lainnya jika diperlukan. Sangat disarankan untuk tidak mengubah-ubah fungsi yang digunakan untuk menyimpan data dari file eksternal ke program karena berpotensi menyebabkan *error* yang tidak diinginkan.
- Daftar perubahan yang terjadi pada *Tablet of Founder* telah disimpan pada *array of struct history*. Sedangkan, daftar mantra *Undounus Incantatem* dan *Redolus Revelio* yang digunakan telah disimpan pada *array of char*. Silahkan periksa *template* program untuk penjelasan lebih lanjut. Disarankan untuk menggunakan Visual Studio Code agar deskripsi fungsi dapat dibaca saat ditunjuk dengan kursor.
- Terdapat tiga file yang digunakan pada program, yakni *main.c* yang berisi program utama, *function_lib.h* yang berisi deklarasi fungsi yang digunakan pada program, serta *function_lib.c* yang berisi implementasi fungsi yang telah dideklarasikan pada *function_lib.h*. Perubahan bebas untuk dilakukan pada file manapun di soal. Gunakan perintah berikut pada terminal secara berurutan untuk melakukan kompilasi:

```
gcc -o main main.c function_lib.c
```

- d. Matriks `tablet_of_founder` merupakan *matrix of character* berukuran 7 x 7 yang merupakan permodelan dari *Tablet of Founder*. Fungsi untuk inisialisasi dan menampilkan *Tablet of Founder* telah dibuatkan pada *template*, silahkan lihat *template* untuk mengetahui cara akses dari matriks `tablet_of_founder`.
- e. Sebelum memulai mengerjakan soal, **coba jalankan program terlebih dahulu** untuk melihat berbagai jenis variabel yang telah dibuatkan pada *template*.
- f. Apabila batas *Undounus Incantatem* atau *Redolus Revelio* tercapai (tidak ada kondisi sebelum atau setelahnya), maka bentuk *Tablet of Founder* tidak mengalami perubahan.

Petunjuk konsep: Dua buah stack (misal stack A dan B) dapat digunakan untuk menyimpan riwayat perubahan yang terjadi karena *Undounus Incantatem* dan *Redolus Revelio*. Stack A digunakan untuk menyimpan perubahan terkini yang terjadi pada *Tablet of Founder*. Pada saat mantra *Undounus Incantatem* diucapkan, maka *head* dari stack A dipindahkan ke stack B. Sebaliknya, saat mantra *Redolus Revelio* diucapkan maka *head* dari stack B dipindahkan kembali ke stack A.

Program menerima sebuah file `.txt`. File berisi daftar perubahan yang terjadi pada *Tablet of Wonder* dari waktu ke waktu serta jumlah dan urutan penggunaan mantra *Undounus Incantatem* atau *Redolus Revelio*. Berikut format dari file `.txt` yang dimaksud.

File Daftar Perubahan ToF

```
<Jumlah penggunaan Undounus Incantatem dan Redolus Revelio>
<Daftar penggunaan Undounus Incantatem atau Redolus Revelio>
<jumlah perubahan>
<indeks baris (int), indeks kolom (int), perubahan akhir (char)>
```

#history1.txt

```
4
U,U,U,R
7
3,3,o // perubahan ke 1
0,0,x // perubahan ke 2
0,6,- // perubahan ke 3
6,0,x // perubahan ke 4
6,6,x // perubahan ke 5
3,3,* // perubahan ke 6
0,6,x // perubahan ke 7
```

#history2.txt

```
5
U,U,R,U,U
9
1,1,o
2,2,o
3,3,o
4,4,o
5,5,o
1,5,o
2,4,o
```

```

4,2,o
5,1,o

#history3.txt
3
U,U,U
1
3,3,o

#history4.txt
10
R,R,R,R,R,R,R,R,R,R
7
3,0,m
3,1,m
3,2,m
3,3,m
3,4,m
3,5,m
3,6,m

```

Contoh Eksekusi Program (garis bawah menandakan *input*):

```

#1
Simulator Sihir Transfigurasi pada Tablet of Founder
Masukkan nama file yang berisi data kondisi mantera pada Tablet of
Founder: history1.txt
Kondisi Tablet of Founder sebelum Undounus Incantatem & Redolus
Revelio
x * * - * * x
* x * - * x *
* * x - x * *
- - - * - - -
* * x - x * *
* x * - * x *
x * * - * * x
Kondisi Akhir Tablet of Founder Setelah Transfigurasi
x * * - * * -
* x * - * x *
* * x - x * *
- - - o - - -
* * x - x * *
* x * - * x *
x * * - * * x

#2
Simulator Sihir Transfigurasi pada Tablet of Founder
Masukkan nama file yang berisi data kondisi mantera pada Tablet of
Founder: history2.txt
Kondisi Tablet of Founder sebelum Undounus Incantatem & Redolus
Revelio

```

```

O * * - * * O
* O * - * O *
* * O - O * *
- - - O - - -
* * O - O * *
* O * - * O *
O * * - * * O

```

Kondisi Akhir Tablet of Founder Setelah Transfigurasi

```

O * * - * * O
* O * - * O *
* * O - X * *
- - - O - - -
* * X - O * *
* X * - * O *
O * * - * * O

```

#3

Simulator Sihir Transfigurasi pada Tablet of Founder

Masukkan nama file yang berisi data kondisi mantera pada Tablet of Founder: history3.txt

Kondisi Tablet of Founder sebelum Undounus Incantatem & Redolus Revelio

```

O * * - * * O
* X * - * X *
* * X - X * *
- - - O - - -
* * X - X * *
* X * - * X *
O * * - * * O

```

Kondisi Akhir Tablet of Founder Setelah Transfigurasi

```

O * * - * * O
* X * - * X *
* * X - X * *
- - - X - - -
* * X - X * *
* X * - * X *
O * * - * * O

```

#4

Simulator Sihir Transfigurasi pada Tablet of Founder

Masukkan nama file yang berisi data kondisi mantera pada Tablet of Founder: history4.txt

Kondisi Tablet of Founder sebelum Undounus Incantatem & Redolus Revelio

```

O * * - * * O
* X * - * X *
* * X - X * *
m m m m m m m
* * X - X * *
* X * - * X *
O * * - * * O

```

Kondisi Akhir Tablet of Founder Setelah Transfigurasi

```
○ * * - * * ○  
* x * - * x *  
* * x - x * *  
m m m m m m m  
* * x - x * *  
* x * - * x *  
○ * * - * * ○
```