



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 : Jumat, 9 April 2021

Naskah Soal Praktikum

Pembuat Naskah: Joshua Adi Chandra, Elang Aditya

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

Seorang mahasiswa ITB bernama Mogu-Mogu saat ini sedang mendata nama-nama orang yang menggunakan aplikasi kalorinya dengan bahasa C. Data tersebut berupa nama, kalori maksimum, dan kalori yang telah diterima dari tubuh.

Program dimulai dengan *input* data dari user dengan jumlah yang tidak ditentukan sebelumnya. Pada proses *input*, data harus divalidasi untuk menyaring data yang tidak benar. Data dikatakan tidak benar apabila kadar kalori maksimum ataupun kalori yang diterima tubuh lebih kecil dari 0. Setelah itu, otomatis program akan mengurutkan data berdasarkan banyaknya jumlah kalori maksimum secara *descending* dan mencetaknya ke layar.

Setelah keluar hasil pengurutan maka user diberikan opsi untuk menghapus sejumlah data dari head (nilai kalori maks yang besar) dengan memasukan jumlah data yang ingin dihapus. Kemudian user akan diminta untuk memasukkan minimum jumlah kalori yang telah diterima tubuh, program kemudian akan menghitung jumlah data dengan jumlah kalori yang diterima tubuh kurang dari sama dengan *input* dan mencetaknya ke layar. Terakhir, user diberikan opsi untuk menambahkan data secara manual pada *tail*.

Pada soal ini kalian dibebaskan dalam membuat fungsi, prosedur, dan lain-lain, namun harus menggunakan *stack* atau *queue* dalam pengerjaannya.

Contoh eksekusi program (garis bawah menunjukkan input)

```
#1
Masukkan nama : George
Masukkan kalori maks : 2000
Masukkan kalori yang diterima : 1000
y untuk lanjut n untuk berhenti : y
Masukkan nama : Edward
Masukkan kalori maks : 2200
Masukkan kalori yang diterima : 1100
y untuk lanjut n untuk berhenti : y
Masukkan nama : Wilfrid
Masukkan kalori maks : 2400
Masukkan kalori yang diterima : 2000
y untuk lanjut n untuk berhenti : y
Masukkan nama : Yoga
Masukkan kalori maks : 2700
Masukkan kalori yang diterima : 1500
y untuk lanjut n untuk berhenti : y
Masukkan nama : Louis
Masukkan kalori maks : 2300
```

Masukkan kalori yang diterima : 2100
y untuk lanjut n untuk berhenti : y
Masukkan nama : Ryan
Masukkan kalori maks : 1900
Masukkan kalori yang diterima : 1000
y untuk lanjut n untuk berhenti : y
Masukkan nama : Hans
Masukkan kalori maks : 2700
Masukkan kalori yang diterima : 1100
y untuk lanjut n untuk berhenti : n

Hans|2700|1100 Ryan|1900|1000 Louis|2300|2100 Yoga|2700|1500
Wilfrid|2400|2000 Edward|2200|1100 George|2000|1000
Hans|2700|1100 Yoga|2700|1500 Wilfrid|2400|2000 Louis|2300|2100
Edward|2200|1100 George|2000|1000 Ryan|1900|1000
Masukkan jumlah data yang ingin dihapus : 2

Wilfrid|2400|2000 Louis|2300|2100 Edward|2200|1100 George|2000|1000
Ryan|1900|1000
Masukkan minimum kalori yang diterima tubuh : 1100

Jumlah user dengan kalori saat ini dibawah 1100 ada sebanyak 3
Masukkan nama : Josh
Masukkan kalori maks : 2200
Masukkan kalori yang diterima : 2000

Wilfrid|2400|2000 Louis|2300|2100 Edward|2200|1100 George|2000|1000
Ryan|1900|1000 Josh|2200|2000

#2

Masukkan nama : Josh
Masukkan kalori maks : -10000
Masukkan kalori yang diterima : 3000
y untuk lanjut n untuk berhenti : y
Masukkan nama : Regina
Masukkan kalori maks : 2200
Masukkan kalori yang diterima : 2000
y untuk lanjut n untuk berhenti : y
Masukkan nama : Ella
Masukkan kalori maks : 2400
Masukkan kalori yang diterima : 1000
y untuk lanjut n untuk berhenti : y
Masukkan nama : Tito
Masukkan kalori maks : 2300
Masukkan kalori yang diterima : 2200
y untuk lanjut n untuk berhenti : y
Masukkan nama : Fadhil
Masukkan kalori maks : 2000

Masukkan kalori yang diterima : 1900
y untuk lanjut n untuk berhenti : n

Fadhil|2000|1900 Tito|2300|2200 Ella|2400|1000 Regina|2200|2000
Ella|2400|1000 Tito|2300|2200 Regina|2200|2000 Fadhil|2000|1900

Masukkan jumlah data yang ingin dihapus : 0

Ella|2400|1000 Tito|2300|2200 Regina|2200|2000 Fadhil|2000|1900
Masukkan minimum kalori yang diterima tubuh : 2000

Jumlah user dengan kalori saat ini dibawah 2000 ada sebanyak 3

Masukkan nama : Hafizh

Masukkan kalori maks : 2300

Masukkan kalori yang diterima : 2000

Ella|2400|1000 Tito|2300|2200 Regina|2200|2000 Fadhil|2000|1900
Hafizh|2300|2000

#3

Masukkan nama : Josh

Masukkan kalori maks : 2200

Masukkan kalori yang diterima : 2100

y untuk lanjut n untuk berhenti : y

Masukkan nama : Tamam

Masukkan kalori maks : 2300

Masukkan kalori yang diterima : 2200

y untuk lanjut n untuk berhenti : y

Masukkan nama : Irvan

Masukkan kalori maks : 2500

Masukkan kalori yang diterima : 2501

y untuk lanjut n untuk berhenti : y

Masukkan nama : Gian

Masukkan kalori maks : 2700

Masukkan kalori yang diterima : 2000

y untuk lanjut n untuk berhenti : n

Gian|2700|2000 Irvan|2500|2501 Tamam|2300|2200 Josh|2200|2100

Gian|2700|2000 Irvan|2500|2501 Tamam|2300|2200 Josh|2200|2100

Masukkan jumlah data yang ingin dihapus : 1

Irvan|2500|2501 Tamam|2300|2200 Josh|2200|2100

Masukkan minimum kalori yang diterima tubuh : 2250

Jumlah user dengan kalori saat ini dibawah 2250 ada sebanyak 2

Masukkan nama : Kevin

Masukkan kalori maks : -2300

Masukkan kalori yang diterima : -1000

Irvan 2500 2501	Tamam 2300 2200	Josh 2200 2100
-----------------	-----------------	----------------

Soal 2

Postfix, *Infix*, dan *Prefix* adalah 3 jenis notasi dalam menulis ekspresi matematis. Perbedaan dari ketiga jenis notasi ini diilustrasikan sebagai berikut:

No.	Infix	Postfix	Prefix
1	$A+B$	$A B +$	$+ A B$
2	$(A + B) * C$	$A B + C *$	$* + A B C$
3	$A * (B + C)$	$A B C + *$	$* A + B C$

Pada notasi *infix*, operator dari suatu ekspresi matematis ditulis antara kedua bilangan yang dioperasikan. Sedangkan pada notasi *postfix* dan notasi *prefix* operator ditulis setelah dan sebelum bilangan-bilangan yang dioperasikan. Notasi *postfix* dan *prefix* memiliki keuntungan karena *order-of-operation* dari notasi ini sudah jelas dimana pada notasi *infix* masih perlu diperhatikan urutan dari tiap operator, seperti mendahulukan perkalian terhadap penjumlahan.

Pak Dodo meminta anda untuk membuat sebuah program yang dapat meng-*convert* suatu ekspresi matematis dengan notasi *infix* menjadi ekspresi matematis dengan notasi *postfix*. Masukkan yang diterima berupa string berisi ekspresi matematis dengan notasi *infix*.

Hint:

- Asumsi input selalu benar
- Gunakan struktur *stack* dalam pembuatan program ini.
- Dapat digunakan converter online untuk memastikan jawaban anda.

Contoh eksekusi program (garis bawah menandakan input)

#1

Masukkan operasi: $A+B*C$

Postfix: $ABC*+$

#2

Masukkan operasi: $A*B+C$

Postfix: $AB*C+$

#3

Masukkan operasi: $(A+B)*(C/D)$

Postfix: $AB+CD/*$

#4

Masukkan operasi: $A*(B*C+D*E)+E$

Postfix: $ABC*DE**F+$

#5

Masukkan operasi: $(A+B)*C+(D-E)/F+G$

Postfix: $AB+C*DE-F/+G+$