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	:	Kamis, 7 April 2022

Naskah Soal Praktikum

Pembuat Naskah: Kevin Naoko, Muhammad Morteza Mudrick

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

Sepasang bilangan dikatakan koprima apabila kedua bilangan tersebut tidak memiliki faktor yang sama selain 1. Tugas anda adalah membuat suatu program yang dapat menerima input beberapa bilangan positif. Program berhenti menerima input apabila user memasukkan bilangan 0. Kemudian, program harus memeriksa sekumpulan angka yang telah dimasukkan oleh user tersebut. Untuk memudahkan penjelasan soal, sekumpulan angka ini akan disebut sebagai "list".

Bila suatu pasangan angka yang bersebelahan (A dan B) pada list tersebut bukan bilangan koprima, maka angka A dan B harus dihilangkan dari list, lalu diganti dengan KPK dari A dan B. Hal ini dilakukan terus menerus hingga semua bilangan yang bersebelahan merupakan koprima. Lebih jelasnya, dapat dilihat contoh iterasi perubahan angka pada Tabel 1

Tabel 1. Contoh perubahan susunan angka ketentuan soal

Iterasi	Isi list	Keterangan
0	[2, 5, 8, 12, 9, 3]	Input awal
1	[2, 5, **8, 12**, 9, 3]	2 dan 5 koprima, dibiarkan 5 dan 8 koprima, dibiarkan 8 dan 12 bukan koprima, replace jadi KPK kedua bilangan (24)
2	[2, 5, **24, 9**, 3]	2 dan 5 koprima, dibiarkan 5 dan 24 koprima, dibiarkan 24 dan 9 bukan koprima, replace jadi KPK kedua bilangan (72)
3	[2, 5, **72, 3**]	2 dan 5 koprima, dibiarkan 5 dan 72 koprima, dibiarkan 72 dan 3 bukan koprima, replace jadi KPK kedua bilangan (72)
4	[2, 5, 72]	Semua bilangan yang bersebelahan sudah koprima, iterasi selesai. List ini merupakan list output

Ketentuan pembuatan program:

- **Praktikan wajib menggunakan tipe data *stack*** dan tidak diperkenankan menggunakan array.
- Pengguna dapat memberikan *input* terus menerus. Bila diberi *input* 0, maka program selesai menerima *input* dan melakukan komputasi.
- Asumsi *input* selalu bilangan bulat positif.

- Fungsi koprima (`coprime`) dan KPK (`lcm`) sudah diberikan di *template* kode. Keterangan *output* untuk fungsi tersebut bisa dilihat pada *template* soal. Praktikan hanya perlu membuat fungsi `push`, `pop`, serta fungsi `main`.
- Gunakan fungsi `printStack` untuk menampilkan *stack input* dan *stack output* agar dapat diperiksa autograder. Sesuaikan pula tipe data yang digunakan fungsi ini dengan tipe data yang anda gunakan.

Contoh Eksekusi Program (garis bawah menandakan input)

```
#1
Input angka ke-1: 2
Input angka ke-2: 5
Input angka ke-3: 8
Input angka ke-4: 12
Input angka ke-5: 9
Input angka ke-6: 3
Input angka ke-7: 0

Stack input: [3, 9, 12, 8, 5, 2]   (terbalik karena stack LIFO)
Stack output: [2, 5, 72]

#2
Input angka ke-1: 15
Input angka ke-2: 36
Input angka ke-3: 11
Input angka ke-4: 18
Input angka ke-5: 27
Input angka ke-6: 45
Input angka ke-7: 11
Input angka ke-8: 16
Input angka ke-9: 28
Input angka ke-10: 0

Stack input: [28, 16, 11, 45, 27, 18, 11, 36, 15]
Stack output: [180, 11, 270, 11, 112]

#3
Input angka ke-1: 3
Input angka ke-2: 5
Input angka ke-3: 15
```

```
Input angka ke-4: 8
Input angka ke-5: 9
Input angka ke-6: 72
Input angka ke-7: 4
Input angka ke-8: 0

Stack input: [4, 72, 9, 8, 15, 5, 3]
Stack output: [360]
```

Soal 2

Salah satu aplikasi *stack* adalah penggunaannya dalam melakukan operasi *undo* (ctrl+Z) dan *redo* (ctrl+Y). Kali ini, anda diminta untuk melakukan pemrograman terkait operasi *undo* dan *redo* pada kalkulator sederhana menggunakan *stack*. Untuk memudahkan pekerjaan kalian, kalkulator hanya akan melakukan operasi penjumlahan

Kalkulator dapat menerima 5 perintah : *undo* (Z), *redo*(Y), cetak (P), penjumlahan(+) dan selesai (F). Perintah *undo* mengembalikan nilai yang disimpan kalkulator menjadi nilai sebelum operasi dilakukan. Perintah *redo* mengembalikan nilai yang disimpan kalkulator menjadi nilai sebelum perintah *undo* dilakukan. Perintah cetak memerintahkan program untuk mencetak nilai terkini. Perintah jumlah (+) diikuti oleh suatu operand. Nilai mula-mula yang disimpan kalkulator adalah 0. Perintah selesai menandakan program sudah selesai menerima *input* dan mencetak nilai akhir.

Apabila perintah *undo* dilakukan, namun tidak ada lagi operasi yang dilakukan sebelumnya (sudah "mentok"), nilai setelah perintah akan sama seperti nilai awal (yaitu 0). Pada kasus ini, cetak "bottom of stack".

Operasi penjumlahan yang dilakukan akan menghapus nilai-nilai *undo* yang disimpan sebelumnya. Oleh karena itu, apabila operasi *redo* dilakukan setelah operasi penjumlahan, *redo* tidak dapat dilakukan dan nilai yang disimpan setelah operasi tersebut bernilai tetap. Pada kasus ini, cetak "top of stack".

Berikut diberikan contoh deretan operasi

Operasi	Nilai yang disimpan setelah operasi	Keterangan	Output Program
	0	Nilai awal	
+ 4	4		
+ 4	8		
P	8	Mencetak nilai	Current Value : 8
Z	4		
Z	0		
Z	0	Sudah mentok	Bottom of stack
Y	4		
+ 3	7		
Y	7	Sudah mentok	Top of stack
+ -7	0		
Z	7		
F	7	Mencetak nilai	Final Value : 7

Format input

Input terdiri dari beberapa baris. Setiap baris dapat berisi 5 tipe operasi (Z,R,P,F,+). Baris yang berisi perintah tambah diikuti dengan operand. Operator dan operand terpisah 1 spasi. Operand dan nilai yang tersimpan **selalu bernilai bilangan bulat**. *Input* selalu diakhiri oleh perintah F.

Format output

Output terdiri dari beberapa baris. *Output* dapat berupa bilangan atau kalimat, tergantung dengan perintah yang diberikan sesuai deskripsi di atas.

Catatan:

- Anda **wajib** menggunakan template operasi pengolahan *stack* serta deklarasi *stack* yang telah disediakan pada file tutorial. Anda diperbolehkan membuat fungsi-fungsi baru dengan dasar fungsi yang diberikan pada file tutorial tersebut
- Diberikan template pembacaan input dan output program
- Anda dapat melakukan pembacaan input dengan sintaks dasar `scanf` (tanpa melakukan tokenisasi). Sintaks `scanf(" %c",…)` membaca 1 karakter saja. Sehingga, ketika terdapat input '+ 4', yang terbaca hanyalah bagian '+' nya saja. Kemudian untuk membaca bagian '4' sebagai *integer*, kalian dapat memberi sintaks `scanf("%d",…)` pada pembacaan berikutnya.

Contoh Eksekusi Program (garis bawah menandakan input)

```
#1
+ 4
+ 4
P
Current Value : 8
Z
Z
Z
Bottom of stack
Y
+ 3
Y
Top of stack
+ -7
Z
F
Final Value : 7
```