

**Topik:** ADT Stack & Queue Dengan List Berkait

**Tujuan Praktikum:**

- Memahami konsep Stack & Queue, terutama dengan implementasi variasi List Linier
- Memahami implementasinya dalam bahasa C.

**PETUNJUK PRAKTIKUM:**

1. Untuk setiap file yang Anda buat, cantumkan header sebagai berikut:

/* NIM>Nama :	
Nama file :	
Topik :	
Tanggal :	
Deskripsi :	*/

2. Seluruh file di-upload setelah dikompres menjadi 1 file dengan nama: P09\_Q<nomor\_soal>\_<nim>.zip.
3. Softcopy materi kuliah/diktat, termasuk yang terkait dengan pemrograman dengan Bahasa C dapat dilihat pada situs <http://kuliah.itb.ac.id/app245/course/> pada link **IF2110 Algorithm & Data Structure**.
4. HANYA ADT YANG DAPAT DI-COMPILE YANG AKAN DIPERIKSA. File yang tidak dapat di-*compile* akan otomatis mendapatkan nilai 0.
5. Tugas ini bersifat INDIVIDUAL. Anda dipersilakan membaca dan menggunakan material yang Anda punya, tetapi tidak ada toleransi bagi pencontek. Jika terbukti, baik yang dicontek maupun yang mencontek akan mendapatkan nilai 0.
6. Ikuti petunjuk asisten untuk pengumpulan tugas ini.

**SELAMAT BEKERJA.**

**Soal 1. Stack****Deskripsi**

Pak Ganesh adalah programmer yang bekerja pada suatu perusahaan IT, ia ingin mengecek kebenaran parenthesis dalam sebuah source code. Parenthesis didefinisikan sebagai **pasangan** ‘{’ dan ‘}’ dimana ‘{’ muncul terlebih dahulu dari ‘}’. Tetapi, diantara parenthesis, bisa saja terdapat body program, yang pada permasalahan ini didefinisikan sebagai “...” (satu atau lebih karakter titik). Misalkan sebuah source code “{....}” maka terdapat 1 pasangan parenthesis. Pada source code “{...{}..}” terdapat 2 pasang parenthesis. Pada source code “{ {} } }” terdapat hanya 2 karakter ‘{’ dan 3 karakter ‘}’, karakter ‘}’ terakhir tidak memiliki parenthesis pembuka sehingga source code program salah. Pada source code “{} } { {” terdapat 2 karakter ‘{’ dan 2 karakter ‘}’, tetapi ‘{’ muncul setelah ‘}’ sehingga source code salah.

Bantulah Pak Ganesh untuk menentukan apakah parenthesis source code program benar atau salah. Buatlah sebuah driver untuk permasalahan ini menggunakan stack yang sudah Anda kerjakan di rumah. Kumpulkanlah (**stack.c** **stack.h** dan **mstack.c**) dalam file **P09\_Q01\_<NIM>.zip**. Anda diperkenankan membuat fungsi/prosedur tambahan bila diperlukan.

**Format Input**

Diberikan sebuah string source code (terdapat newline di akhir). Maksimal berisi 30 karakter. Cara membaca:

```
while (scanf("%c", &ch)==1) {
    //body
    //ch adalah karakter per karakter dari string input
}
```

**Format Output**

Outputkan “ok” apabila parenthesis pada source code benar atau “salah” apabila parenthesis pada source code salah. Jangan lupa untuk memberikan newline.

**Contoh Input-Output**

Input	Output
{...}	ok
{...{}...}	ok
{{{}}...	salah
{{...}}	salah
{ } { } . . { ... }	salah
} . . { . . } { . . }	salah
{ ... } { ... }	salah

Untuk memudahkan pengerjaan, Anda dapat men-*supply* input dari file. Misal Anda men-*supply* input dari file *input.in* dan nama object file hasil kompilasi adalah *mstack*, maka Anda dapat run dengan cara “./mstack < *input.in*”.

Input.in berisi string input seperti pada contoh input-output, misal

```
{...}
```

**Soal 2. Queue (Priority Queue)****Deskripsi**

Bu Dengklek adalah guru matematika yang sadis. Karena Bu Dengklek merupakan guru yang dihormati oleh semua orang di sekolah, semua peraturan dari Bu Dengklek akan dituruti, termasuk peraturan antrian kantin. Pada saat bel makan siang berbunyi, semua siswa akan serta merta pergi ke kantin untuk makan siang. Pada kantin sekolah, terdapat peraturan khusus yakni urutan antrian pada kantin ditentukan berdasarkan nilai ujian matematika terakhir siswa!. Siswa yang memiliki nilai matematika lebih besar boleh menyerobot antrian, **kecuali siswa terdepan (karena makanannya sudah sedang dipersiapkan)**. Untuk setiap siswa yang mengantri, Ibu kantin akan mempersiapkan makanan satu per satu untuk siswa terdepan antrian. Untuk setiap siswa, diketahui nilai matematika, waktu yang dibutuhkan ibu kantin untuk mempersiapkan makanannya serta waktu kedatangan siswa di kantin. Karena hanya ada 1 ibu kantin yang mempersiapkan makanan, hanya terdapat 1 antrian pada kantin sekolah.

Lihatlah tabel berikut ini ini

Nama mahasiswa	Nilai Matematika	Waktu persiapan makanan	Waktu kedatangan
Melvin	80	3	1
Michi	90	1	2
Wira	99	2	3

Urutan kedatangan adalah Melvin, Michi kemudian Wira. Saat Melvin datang, dibutuhkan waktu persiapan makanan untuk Melvin sebesar 3 satuan waktu. Saat waktu menunjukkan 2, Michi datang, sementara itu makanan Melvin belum selesai dipersiapkan. Saat waktu baru menunjukkan 3, Wira datang menyerobot, kemudian Michi menjadi berada pada posisi antrian ke-3, pada akhir waktu ke-3 makanan Melvin sudah selesai dipersiapkan, sehingga Wira yang akan dilayani berikutnya (bukan Michi, ingat kantin ini bekerja sesuai dengan aturan Bu Dengklek. Wira tidak dapat menyerobot Melvin karena sudah keluar dari antrian dan sedang dilayani). Kemudian Ibu kantin akan mempersiapkan makanan Wira sebesar 2 satuan waktu. Sehingga, Michi membutuhkan total **waktu tunggu hingga mulai dilayani** di kantin sebesar 4 satuan waktu semenjak kedatangannya sampai makanannya mulai dipersiapkan, sedangkan Wira membutuhkan waktu tunggu sebesar 1 satuan waktu dan Melvin hanya membutuhkan waktu tunggu sebesar 0. Ingat, **siswa paling depan sudah mulai dilayani, sehingga tidak dapat diserobot**.

Perhatikan tabel berikut sebagai ilustrasi antrian pelayanan pada deskripsi.

Waktu 1	Waktu 2	Waktu 3	Waktu 4	Waktu 5	Waktu 6
[Melvin]	[Melvin, Michi]	[Melvin, Wira, Michi]	[Wira, Michi]	[Wira, Michi]	[Michi]

Simulasikanlah antrian kantin seperti pada definisi. Kumpulkanlah (**queue.c** **queue.h** dan **mqueue.c**) dalam file **P09\_Q02\_<NIM>.zip**. Anda diharuskan mengubah ADT yang sudah dikerjakan menjadi priority queue yang disesuaikan dengan soal, Anda diperkenankan membuat fungsi/prosedur tambahan bila diperlukan.

**Format Input**

- Baris pertama berisi sebuah bilangan N yang menandakan banyaknya siswa
- Baris kedua sampai N+1 berisi 3 buah bilangan yaitu M, T dan A yang menandakan Nilai Matematika, Waktu persiapan makanan serta waktu kedatangan siswa ke-i.

**Constraint**

- Nilai matematika diantara siswa tidak dijamin berbeda, bila 2 siswa memiliki nilai matematika yang sama maka diurutkan berdasarkan waktu kedatangan. Waktu kedatangan dijamin unik.
- N, M, T, A adalah bilangan bulat.  $0 \leq N, T, A \leq 20$ .  $0 \leq M \leq 100$ .
- Input dipastikan terurut sesuai dengan waktu kedatangan siswa.
- Siswa terdepan diasumsikan sudah keluar dari antrian karena sedang dilayani ibu kantin.

Berikut kami berikan opsi perubahan Infotype. Silahkan gunakan sesuai dengan kebutuhan.

```
typedef struct {
    int id; //urutan kedatangan siswa di kantin "siswa ke-i"
    int nilaiMath; //M
    int lamaPelayanan; //T, persiapan makanan untuk siswa
    int waktuKedatangan; //A
    int mulaiPelayanan; //waktu persiapan makanan untuk siswa dimulai
} Infotype;
```

**Format Output**

Berisi N buah baris, tiap baris berisi <i, W> dimana i merupakan urutan kedatangan dan W merupakan total waktu tunggu di kantin untuk siswa ke-i. Outputkanlah sesuai dengan urutan keluar siswa dari antrian. Nomor siswa diasumsikan mulai dari 1.

**Contoh Input-Output**

Input	Output
3 80 3 1 90 1 2 99 2 3	1 0 3 1 2 4
2 90 1 1 99 2 2	1 0 2 0
3 80 11 1 92 10 9 99 2 10	1 0 3 2 2 5
3 80 2 10 90 1 11 99 2 20	1 0 2 1 3 0
5 70 5 1 80 5 8 90 2 9 90 1 10 99 2 20	1 0 2 0 3 4 4 5 5 0
3 80 5 1 80 3 2 80 1 3	1 0 2 4 3 6

Simulasi antrian pada contoh input-output terakhir

```
3
80 5 1 //siswa1
80 3 2 //siswa2
80 1 3 //siswa3
```

Waktu 1	Waktu 2	Waktu 3	Waktu 4	Waktu 5	Waktu 6	Waktu 7	Waktu 8	Waktu 9
Siswa1	Siswa1, Siswa2	Siswa1, Siswa2, Siswa3	Siswa1, Siswa2, Siswa3	Siswa1, Siswa2, Siswa3	Siswa2, Siswa3	Siswa2, Siswa3	Siswa2, Siswa3	Siswa3