

Started on	Monday, 14 November 2022, 7:05 PM
State	Finished
Completed on	Monday, 14 November 2022, 7:10 PM
Time taken	4 mins 52 secs
Marks	750.00/750.00
Grade	100.00 out of 100.00

Question **1**  
 Correct  
 Mark 160.00 out of 160.00

Time limit	1 s
Memory limit	64 MB

Kumpulkan **queuelinked.c**



[queuelinked.c](#)

Score: 160

Blackbox

Score: 160

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.64 MB
2	10	Accepted	0.00 sec, 1.71 MB
3	10	Accepted	0.00 sec, 1.64 MB
4	10	Accepted	0.00 sec, 1.63 MB
5	10	Accepted	0.00 sec, 1.64 MB
6	10	Accepted	0.00 sec, 1.67 MB
7	10	Accepted	0.00 sec, 1.64 MB
8	10	Accepted	0.00 sec, 1.64 MB
9	10	Accepted	0.00 sec, 1.63 MB
10	10	Accepted	0.00 sec, 1.59 MB
11	10	Accepted	0.00 sec, 1.71 MB
12	10	Accepted	0.00 sec, 1.71 MB
13	10	Accepted	0.00 sec, 1.66 MB
14	10	Accepted	0.00 sec, 1.60 MB
15	10	Accepted	0.00 sec, 1.55 MB
16	10	Accepted	0.00 sec, 1.62 MB

Question **2**

Correct

Mark 140.00 out of 140.00

Time limit	1 s
Memory limit	64 MB

Kumpulkan **stacklinked.c**



[stacklinked.c](#)

Score: 140

Blackbox

Score: 140

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.60 MB
2	10	Accepted	0.00 sec, 1.51 MB
3	10	Accepted	0.00 sec, 1.66 MB
4	10	Accepted	0.00 sec, 1.64 MB
5	10	Accepted	0.00 sec, 1.58 MB
6	10	Accepted	0.00 sec, 1.50 MB
7	10	Accepted	0.00 sec, 1.67 MB
8	10	Accepted	0.00 sec, 1.66 MB
9	10	Accepted	0.00 sec, 1.66 MB
10	10	Accepted	0.00 sec, 1.62 MB
11	10	Accepted	0.00 sec, 1.60 MB
12	10	Accepted	0.00 sec, 1.60 MB
13	10	Accepted	0.00 sec, 1.64 MB
14	10	Accepted	0.00 sec, 1.54 MB

Question **3**

Correct

Mark 200.00 out of 200.00

Time limit	1 s
Memory limit	8 MB

**Parser Kurung**

Kakek Tuan Zaki sangat senang membuat soal permainan kurung dan mewajibkan cucunya untuk mengerjakannya. Namun, karena Tuan Zaki memiliki pekerjaan lain dan menjadi sangat sibuk ia menjadi terlalu lelah serta tidak sempat berpikir untuk mengerjakan soal kakeknya. Sehingga, ia ingin membuat program untuk menjawab persoalan tersebut. Persoalan ini cocok diselesaikan dengan struktur data **stack**. Bantulah Kakek Tuan Zaki untuk mengecek apakah sebuah kalimat memiliki kurung yang valid atau tidak.

Aturan dan atau Batasan:

- Diberikan **string** yang diakhiri dengan **mark** titik (.), Kalimat mungkin kosong.
- Diharuskan menggunakan **stack** variasi linked list dan **mesin karakter**
- Kalimat kurung terdiri dari **karakter apapun**, namun yang harus dicek adalah **tanda kurung**. Abaikan karakter lain.
- Setiap menemukan **kurung pembuka**, lakukan **push**, setiap menemukan **kurung penutup**, lakukan **pop**.
- Setiap menemukan kurung buka, tutup, dan tegak, **display stack**.
- Apabila sudah terbaca semua karakternya dan kurung valid, print "kurung valid" dan "[A] (B) {C} {D} <E>", dengan **A** jumlah pasangan kurung siku, **B** jumlah pasangan kurung bulat, dst. Kemudian, print "**MAX F**", dengan **F** jumlah maksimum panjang stack.
- Jika tidak valid, print "kurung tidak valid".
- Aturan kurung menggunakan aturan kurung matematika, sehingga jika berpasangan namun urutan tidak benar, menjadi tidak valid. Contoh: [( )] adalah kurung yang tidak valid. Jika tidak ada kurung yang menyalahi aturan, maka kalimat dianggap valid.
- Kumpulkan **stacklinked.c** dan **parserkurung.c** yang di zip menjadi **parserkurung.zip**
- Gunakan integer untuk stack anda, lihat tabel berikut untuk informasi representasi.

Tanda Kurung	Representasi Integer
Kurung Siku [ ]	0
Kurung Bulat ( )	1
Kurung Tegak	2
Kurung Kurawal { }	3
Kurung Segitiga < >	4

- Untuk kurung tegak ||, apabila berurut berkali-kali, contoh: ||||, diprioritaskan push pop push pop, **bukan push push pop pop**. Namun jika kasusnya, [(||)], maka akan terjadi push push push pop pop pop.

**Hint:** Hati-hati dengan Kurung Tegak || karena dua-duanya merupakan karakter yang sama.

Berikut adalah contoh input dan output.

Input	Output	Keterangan
<5,2> dan <7,3> adalah koordinat rahasia dan disana ada agen dengan id {1,8,2,1,3,5}.	[4] [] [4] [] [3] [] kurung valid [0] (0)  0  {1} <2> MAX 1	Saat pertama kali menemukan karakter kurung <, push lalu display stack.  Lalu menemukan >, pop, lalu display stack.  Begitu seterusnya. Panjang Stack maksimum 1.  Jangan lupa, cetak jumlah pasangan kurung.
$[(1+1)*(7/3)-(6/3+-2*9)] = x$ .	[0] [1,0] [0] [1,0] [0] [2,0] [1,2,0] [2,0] [0] [] kurung valid [1] (3)     {0} <0> MAX 3	<b>Ingat!</b> Hanya cetak stack ketika bertemu dengan kurung buka/kurung tutup/kurung tegak. Huruf lain diabaikan.
[f{e<a>d>c}].	[0] [3,0] [4,3,0] [4,3,0] [3,0] [3,0] kurung tidak valid	Kurung berpasangan tetapi tidak valid karena penutupan kurung tidak benar.
(: semoga harimu menyenangkan.	[1] kurung tidak valid	Kurung buka = push
:) semoga harimu menyenangkan.	[] kurung tidak valid	Kurung tutup = pop
Hai.	kurung valid [0] (0)  0  {0} <0> MAX 0	Tidak ada kurung sehingga tidak perlu display stack. Kurung valid karena kalimat tidak menyalahi aturan kurung.



 [parserkurung.zip](#)

Score: 200

Blackbox

Score: 200

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	5	Accepted	0.00 sec, 1.58 MB
2	5	Accepted	0.00 sec, 1.47 MB
3	5	Accepted	0.00 sec, 1.60 MB
4	5	Accepted	0.00 sec, 1.47 MB
5	5	Accepted	0.00 sec, 1.58 MB
6	5	Accepted	0.00 sec, 1.56 MB
7	5	Accepted	0.00 sec, 1.47 MB

No	Score	Verdict	Description
8	10	Accepted	0.00 sec, 1.58 MB

9	10	Accepted	0.00 sec, 1.65 MB
10	10	Accepted	0.00 sec, 1.59 MB
11	15	Accepted	0.00 sec, 1.57 MB
12	30	Accepted	0.00 sec, 1.46 MB
13	20	Accepted	0.00 sec, 1.54 MB
14	20	Accepted	0.00 sec, 1.60 MB
15	10	Accepted	0.00 sec, 1.64 MB
16	5	Accepted	0.00 sec, 1.59 MB
17	5	Accepted	0.00 sec, 1.53 MB
18	5	Accepted	0.00 sec, 1.60 MB
19	15	Accepted	0.00 sec, 1.52 MB
20	10	Accepted	0.00 sec, 1.54 MB

Question **4**

Correct

Mark 250.00 out of 250.00

Time limit	1 s
Memory limit	8 MB

**Sandwich Butak**

Di sebuah Asrama Latak (buLAT koTAK), mahasiswa yang tinggal di sana diberi makan siang gratis setiap harinya. Makanan siang tersebut selalu sandwich yang dinamakan Butak (BUlat koTAK) karena sandwichnya ada yang berbentuk bulat dan atau kotak. Setiap jam makan siang, semua mahasiswa mengantri. Namun, setiap mahasiswa memiliki preferensinya masing-masing, sehingga jika bentuk sandwich yang ada tidak sesuai, mahasiswa tersebut akan keluar dari antrian lalu mengantri kembali. Sebaliknya, jika sesuai keinginan, mahasiswa akan keluar dari antrian dan mengambil sandwich tersebut. Bantu petugas asrama menghitung jumlah mahasiswa yang tidak akan dapat sandwich karena preferensinya tidak sesuai.

Aturan dan atau Batasan:

- **Jumlah sandwich selalu sama dengan jumlah mahasiswa.** Dan **tidak mungkin string kosong.**
- **Setiap mahasiswa memiliki nomor.** Nomor ini berurut dari 1 hingga N dengan N jumlah mahasiswa.
- Diharuskan menggunakan **queue variasi linked list dan mesin karakter.**
- Input dipastikan valid. Dan hanya terdiri dari huruf B dan K.
- Input dimulai dengan antrian mahasiswa lalu diakhiri dengan **mark koma (,)** Lalu dilanjutkan dengan sandwich yang tersedia lalu diakhiri dengan **mark titik (.)**
- Setiap ada mahasiswa yang mendapat sandwich, cetak "**A -> B**" dengan **A** adalah nomor mahasiswa, **B** adalah bulat/kotak.
- Diakhir print "**X**", X = **jumlah mahasiswa yang tidak dapat sandwich.**
- Gunakan integer untuk **queue sandwich & mahasiswa.** Representasi: 0 untuk Bulat dan 1 untuk Kotak.
- Untuk **queue nomor mahasiswa**, isi 1 sampai N (jumlah mahasiswa pada mula-mula).
- Kumpulkan **queuelinked.c** dan **butak.c** yang di zip menjadi **butak.zip**

Berikut adalah contoh input dan output.

Input	Output	Keterangan
BBKK,KBKB.	3 -> kotak 1 -> bulat 4 -> kotak 2 -> bulat 0	<p>Kondisi Awal: Mahasiswa: [0,0,1,1] -&gt; BBKK Nomor Mahasiswa: [1,2,3,4] -&gt; awalnya dibuat 1 sampai N berurut Sandwich: [1,0,1,0] -&gt; KBKB</p> <p>Mahasiswa Nomor 1 maunya bulat, tetapi sandwich adanya kotak. Mengantri ulang ke belakang. (dequeue, lalu enqueue) Mahasiswa: [0,1,1,0] Nomor Mahasiswa: [2,3,4,1] Sandwich: [1,0,1,0] (queue sandwich hanya boleh di dequeue ketika sandwich diambil sesuai keinginan mahasiswa)</p> <p>Kasus yang sama dengan mahasiswa nomor 2, mengantri ulang. Mahasiswa: [1,1,0,0] Nomor Mahasiswa: [3,4,1,2] Sandwich: [1,0,1,0]</p> <p>Mahasiswa nomor 3 mendapat apa yang ia mau. 3 -&gt; kotak Mahasiswa: [1,0,0] Nomor Mahasiswa: [4,1,2] Sandwich: [0,1,0]</p> <p>Mahasiswa nomor 4 mengantri ulang karena tidak sesuai. Sandwich adanya bulat, ia maunya kotak. Mahasiswa: [0,0,1] Nomor Mahasiswa: [1,2,4] Sandwich: [0,1,0]</p> <p>Mahasiswa nomor 1 mendapat apa yang ia mau. 1 -&gt; bulat Mahasiswa: [0,1] Nomor Mahasiswa: [2,4] Sandwich: [1,0]</p> <p>Nomor 2 mengantri ulang karena tidak sesuai. Mahasiswa: [1,0] Nomor Mahasiswa: [4,2] Sandwich: [1,0]</p> <p>Nomor 4 mendapat apa yang ia mau. 4 -&gt; kotak Mahasiswa: [0] Nomor Mahasiswa: [2] Sandwich: [0]</p> <p>Nomor 2 mendapat apa yang ia mau. 2 -&gt; bulat</p> <p>Kondisi Akhir. Semua dapat sandwich. Mahasiswa: [] Nomor Mahasiswa: [] Sandwich: [] 0</p>

BBK,KKK.	3 → kotak 2	<p>Kondisi Awal: Mahasiswa: [0,0,1] → BBK Nomor Mahasiswa: [1,2,3] Sandwich: [1,1,1] → KKK</p> <p>Mahasiswa Nomor 1 ngantri ulang. Mahasiswa: [0,1,0] Nomor Mahasiswa: [2,3,1] Sandwich: [1,1,1]</p> <p>Mahasiswa Nomor 2 ngantri ulang. Mahasiswa: [1,0,0] Nomor Mahasiswa: [3,1,2] Sandwich: [1,1,1]</p> <p>Mahasiswa nomor 3 dapat apa yang ia mau. 3 → kotak</p> <p>Kondisi Akhir: Mahasiswa: [0,0] Nomor Mahasiswa: [1,2] Sandwich: [1,1]</p> <p>Diakhir ada 2 orang yang tidak dapat sandwich.</p>
BBKB,KKBB.	3 → kotak 3	<p>Yang dapat cuman nomor 3, karena sandwichnya tertahan dengan bentuk kotak dan tidak ada yang mau mengambil.</p> <p>Kondisi Akhir: Mahasiswa: [0,0,0] Nomor Mahasiswa: [4,1,2] Sandwich: [1,0,0] Diakhir 3 orang tidak dapat sandwich.</p>
KKKBK,KB BBK.	1 → kotak 4 → bulat 5 → bulat 3	3 orang tidak dapat sandwich.

**Hint:** buatlah fungsi `isElHomogen(Queue q)` untuk mengecek apakah suatu queue isinya adalah elemen yang sama. Jangan gunakan fungsi `dequeue` pada fungsi ini! Fungsi ini mirip cara kerjanya dengan `display queue`: iterasi elemen tanpa `dequeue`. Fungsi ini digunakan untuk menyelesaikan loop, jika queue mahasiswa sudah tidak bisa mengambil sandwich lagi (contoh: Queue Mahasiswa homogen bernilai 0, namun, head queue sandwich nilainya 1, maka keluar dari loop).



 [butak.zip](#)

Score: 250

Blackbox

Score: 250

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	5	Accepted	0.00 sec, 1.55 MB
2	5	Accepted	0.00 sec, 1.55 MB
3	5	Accepted	0.00 sec, 1.55 MB
4	5	Accepted	0.00 sec, 1.56 MB
5	5	Accepted	0.00 sec, 1.56 MB
6	5	Accepted	0.00 sec, 1.52 MB
7	5	Accepted	0.00 sec, 1.57 MB
8	5	Accepted	0.00 sec, 1.58 MB



No	Score	Verdict	Description
	5	Accepted	0.00 sec, 1.55 MB

10	10	Accepted	0.01 sec, 1.66 MB
11	10	Accepted	0.00 sec, 1.55 MB
12	5	Accepted	0.00 sec, 1.59 MB
13	5	Accepted	0.00 sec, 1.56 MB
14	5	Accepted	0.00 sec, 1.53 MB
15	5	Accepted	0.00 sec, 1.58 MB
16	5	Accepted	0.00 sec, 1.46 MB
17	5	Accepted	0.00 sec, 1.46 MB
18	5	Accepted	0.00 sec, 1.58 MB
19	10	Accepted	0.00 sec, 1.62 MB
20	10	Accepted	0.00 sec, 1.61 MB
21	30	Accepted	0.00 sec, 1.60 MB
22	30	Accepted	0.01 sec, 1.68 MB
23	20	Accepted	0.00 sec, 1.59 MB
24	25	Accepted	0.00 sec, 1.55 MB
25	25	Accepted	0.00 sec, 1.55 MB