

Question **4**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Implementasikan [removeDupes.h](#) kedalam file removeDupes.c

Gunakan ADT [queue.c](#) dan [queue.h](#) sebagai bantuan dalam menyelesaikan soal.

[boolean.h](#)


C

⬆

⬇

⬆

⬇

 [removeDupes.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.62 MB
2	10	Accepted	0.00 sec, 1.55 MB
3	10	Accepted	0.00 sec, 1.48 MB
4	10	Accepted	0.00 sec, 1.59 MB
5	10	Accepted	0.00 sec, 1.55 MB
6	10	Accepted	0.00 sec, 1.54 MB
7	10	Accepted	0.00 sec, 1.67 MB
8	10	Accepted	0.00 sec, 1.60 MB
9	10	Accepted	0.00 sec, 1.50 MB
10	10	Accepted	0.00 sec, 1.59 MB

Time limit	1 s
Memory limit	64 MB

Durasi Antre

Buat program dengan nama **durasiantre.c** untuk mengimplementasikan [durasiantre.h](#).

Gunakan [queue.h](#) dan [queue.c](#) dan [boolean.h](#)

Deskripsi

Pada praktikum sebelumnya, Anda pernah diminta membantu memperbaiki mesin antre **Labprosotan Theme Park**. Kini, taman bermain tersebut memiliki masalah baru: munculnya calo tiket masuk!

Labprosotan Theme Park hanya memperbolehkan satu pembelian tiket per orang per pembelian untuk mencegah calo tiket. Artinya, jika Anda ingin membeli 3 tiket, maka Anda harus mengantre sebanyak tiga kali. Setiap pembelian tiket membutuhkan waktu 1 satuan waktu.

Artinya jika terdapat antrian [2, 2, 3] dan anda berada pada posisi indeks ke-1, maka antrian ke-0 akan diproses. Karena belum terbeli seluruh tiketnya, elemen tersebut akan mengantri kembali. Kini antrian merupakan [2, 3, 1] dengan anda berada di posisi indeks ke-0. Pemrosesan ini terus diulang sampai Anda telah membeli seluruh tiket.

Anda diminta oleh seorang calo tiket untuk membuat program yang dapat menghitung berapa total waktu yang diperlukan seseorang untuk membeli seluruh tiket yang diperlukan, baik waktu membeli tiketnya maupun menunggu antrean.

Format Masukan

Sebuah *queue* berisi integer dan sebuah integer.

Format Keluaran

Sebuah integer.

Batasan

Queue awal tidak kosong, elemen pada *queue* awal tidak mungkin nol.

Contoh Input 1

Masukan:

Q = [3,4,2,4,6]

k = 2

Keluaran:

8

Penjelasan

Awal antrian: [3, 4, 2, 4, 6]
^ anda

Iterasi-1: Elemen 3 akan dikurang dan mengantri kembali
Isi antrian setelah: [4, 2, 4, 6, 2]
^ anda

Iterasi-2: Elemen 3 akan dikurang dan mengantri kembali
Isi antrian setelah: [2, 4, 6, 2, 3]
^ anda

Iterasi-3: Elemen 3 akan dikurang dan mengantri kembali
Isi antrian setelah: [4, 6, 2, 3, 1]
^ anda

Iterasi-4: Elemen 3 akan dikurang dan mengantri kembali
Isi antrian setelah: [6, 2, 3, 1, 3]
^ anda

Iterasi-5: Elemen 3 akan dikurang dan mengantri kembali
Isi antrian setelah: [2, 3, 1, 3, 5]
^ anda

Iterasi-6: Elemen 3 akan dikurang dan mengantri kembali
Isi antrian setelah: [3, 1, 3, 5, 1]
^ anda

Iterasi-7: Elemen 3 akan dikurang dan mengantri kembali
Isi antrian setelah: [1, 3, 5, 1, 2]
^ anda

Iterasi-8: Elemen 1 akan dikurang dan seluruh tiket sudah dibeli
Isi antrian setelah: [3, 5, 1, 2]

Sehingga total waktu mengantri adalah 8

Contoh Input 2

Masukan:

Q = [1,2]
k = 1

Keluaran:

3

Penjelasan

Awal antrian: [1, 2]
^ anda

Iterasi-1: Elemen 1 akan dikurang dan habis
Isi antrian setelah: [2]
^ anda

Iterasi-2: Elemen 2 akan dikurang dan mengantri kembali
Isi antrian setelah: [1]
^ anda

Iterasi-3: Elemen 1 akan dikurang dan seluruh tiket sudah dibeli
Isi antrian setelah: []

Sehingga total waktu mengantri adalah 3

Hint

- Selalu simpan berapa banyak tiket anda sudah dibeli
- Tiket yang sudah habis dibeli (anda atau lain) tidak dimasukan lagi ke queue

C

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.65 MB
2	10	Accepted	0.00 sec, 1.56 MB
3	10	Accepted	0.00 sec, 1.62 MB
4	10	Accepted	0.00 sec, 1.60 MB
5	10	Accepted	0.00 sec, 1.71 MB
6	10	Accepted	0.00 sec, 1.63 MB
7	10	Accepted	0.00 sec, 1.64 MB
8	10	Accepted	0.00 sec, 1.49 MB
9	10	Accepted	0.00 sec, 1.55 MB
10	10	Accepted	0.00 sec, 1.54 MB

Time limit	1 s
Memory limit	64 MB

Ibu Lia adalah pemilik dari restoran bintang 5, kueue, dan terkadang datang ke restoran tersebut untuk melihat kondisi di lapangan dan membantu memasak. Tidak seperti para koki yang lainnya, Ibu Lia hanya dapat memasak 1 masakan untuk 1 waktunya, dengan setiap masakannya membutuhkan waktu yang berbeda-beda, dan tamu-tamu lainnya dapat melakukan permintaan untuk mendapat masakan dari Ibu Lia, karena masakannya yang sangat populer dan juga viral di social media. Suatu hari, Ibu Lia ingin mengetahui berapa jumlah masakan yang telah disajikan, waktu minimal yang dibutuhkan, dan waktu maksimal yang dibutuhkan pada 1 hari tertentu. Menurut Ibu Lia, masalah ini dapat diselesaikan dengan membuat kode dalam bahasa C.

[queue.h](#)

[queue.c](#)

[boolean.h](#)

Spesifikasi program:

1. Input berupa angka, yaitu 0, 1, dan 2. 0 berarti program berhenti menerima masukan. 1 berarti menerima permintaan (dengan tambahan parameter adalah waktu yang dibutuhkan dan waktu yang tidak valid akan diabaikan). 2 berarti menyajikan masakan.
2. Output berupa jumlah masakan yang disaji, waktu minimal yang dibutuhkan, dan waktu maksimal yang dibutuhkan (diikuti dengan newline). Default value adalah 0.
3. Nama file adalah **restoran.c**.

Berikut adalah contoh interaksi input/output program.

Input	Output	Keterangan
1 2 1 5 2 0	1 2 2	Karena hanya menyajikan 1 masakan, maka total adalah 1, waktu minimal dan waktu maksimal yang dibutuhkan adalah 2
1 2 2 1 5 2 0	2 2 2 5	Karena menyajikan 2 masakan, maka total adalah 2, dan waktu minimal yang dibutuhkan adalah 2 (pada pesanan pertama) dan waktu maksimal yang dibutuhkan adalah 5(pada pesanan kedua)

C

 [restoran.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 1.55 MB
2	20	Accepted	0.00 sec, 1.63 MB
3	20	Accepted	0.00 sec, 1.66 MB

No	Score	Verdict	Description
4	20	Accepted	0.00 sec, 1.55 MB
5	20	Accepted	0.00 sec, 1.64 MB

[◀ Praktikum 6](#)

Jump to...

⌵

[Feedback Form Praktikum 6 ▶](#)