

[Dashboard](#) / [My courses](#) / [ITB IF2110 1 2324](#) / [Praktikum 7 \(16 October - 22 October\)](#) / [ADT Queue - Praktikum \(extended\)](#)

Started on	Thursday, 30 November 2023, 6:40 PM
State	Finished
Completed on	Thursday, 30 November 2023, 7:20 PM
Time taken	40 mins 41 secs
Grade	60.00 out of 120.00 (50%)

Question **1**

Correct

Mark 30.00 out of 30.00

Time limit	1 s
Memory limit	64 MB

Pada saat-saat tertentu, kantin Labtek V selalu penuh oleh mahasiswa yang mengantre ingin membeli makanan, termasuk Tuan Man. Suatu hari, Tuan Man penasaran dengan lama waktu antrean rata-rata semua orang. Tuan Man menyadari antrean tersebut mirip dengan Queue yang ia pelajari di Mata Kuliah Alstrukdat saat ini. Bantulah Tuan Man dalam menghitung lama waktu rata-rata antrean mahasiswa yang sudah dilayani.

Buatlah sebuah program utama yang menggunakan ADT Queue yang sudah Anda kerjakan sebagai tugas pra-praktikum untuk mensimulasikan antrean kantin Labtek V. Program tersebut diberi nama **antreankantin.c**. Elemen Queue menyatakan lama waktu mahasiswa. Estimasi lama waktu mengantre berkisar antara 1 hingga 20 menit (inklusif). Kapasitas dari queue sesuai dengan header = 100.

Program harus melakukan hal-hal sebagai berikut:

Program dimulai dengan loop. Pengguna dapat memilih untuk menambah atau menghapus antrean 1 mahasiswa (menghapus antrean berarti mahasiswa tersebut sudah dilayani). Pengguna memiliki 3 opsi, yaitu tambah antrean, hapus antrean, dan penghentian proses. Proses penambahan dan penghapusan antrean mengikuti ketentuan operasi enqueue dan dequeue.

1. Tambah antrean. Kode operasi : 1 kemudian diikuti dengan estimasi lama waktu mengantre. Contoh: 1 5 (menambah sebuah elemen queue dengan estimasi lama waktu antrean 5 menit). Jika ingin menambahkan antrean namun queue sudah penuh, tampilkan pesan "Queue penuh".
2. Hapus antrean. Kode operasi : 2. Jika ingin menghapus antrean namun queue kosong, tampilkan pesan "Queue kosong".
3. Penghentian proses. Kode operasi = 0.

Di akhir program, tampilkan jumlah mahasiswa yang dilayani pada baris pertama dan waktu rata-rata antrean yang sudah dilayani dalam format 2 angka di belakang koma pada baris kedua. Apabila belum ada mahasiswa yang dilayani, nilai rata-rata digantikan dengan pesan "Tidak bisa dihitung"

Petunjuk:

Untuk mengeluarkan nilai desimal/float dengan 2 angka di belakang koma, gunakan kode program berikut

printf("%.2f", var)

Format masukan

$C_i X_i$

Catatan: C bernilai 0, 1, atau 2 sesuai kode operasi di atas. Jika C bernilai 1, X bernilai 1 - 20

Format keluaran

Pesan Queue penuh/kosong

N (jumlah mahasiswa yang telah dilayani)

X (rata-rata lama waktu antrean dalam format 2 angka di belakang koma)

Berikut adalah contoh interaksi input/output program. Input program ditandai dengan garis bawah

Masukan dan keluaran	Keterangan
<u>1</u> <u>3</u>	Jumlah mahasiswa yang telah dilayani = 0 Waktu rata-rata tidak dapat dihitung
<u>1</u> <u>4</u>	
<u>0</u>	
<u>0</u>	
Tidak bisa dihitung	

1 3 1 4 2 0 1 3.00	Jumlah mahasiswa yang telah dilayani = 1 Waktu rata-rata: $3/1 = 3$
2 Queue kosong 1 3 1 4 2 2 0 2 3.50	Jumlah mahasiswa yang telah dilayani = 2 Waktu rata-rata: $(3+4)/2 = 3.50$
... 1 5 1 9 1 10 1 10 Queue penuh 1 9 Queue penuh 2 2 2 ...	Kasus Queue penuh

C

 [antreankantin.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.54 MB
2	10	Accepted	0.00 sec, 1.45 MB
3	10	Accepted	0.00 sec, 1.51 MB
4	10	Accepted	0.00 sec, 1.57 MB
5	10	Accepted	0.00 sec, 1.49 MB
6	10	Accepted	0.00 sec, 1.50 MB

No	Score	Verdict	Description
7	10	Accepted	0.00 sec, 1.47 MB
8	10	Accepted	0.00 sec, 1.45 MB
9	10	Accepted	0.01 sec, 1.50 MB
10	10	Accepted	0.00 sec, 1.51 MB

Question **2**

Correct

Mark 30.00 out of 30.00

Time limit	1 s
Memory limit	64 MB

Priority Queue adalah struktur antrian dengan prioritas tertentu sehingga elemen-elemen queue disusun terurut berdasarkan prioritas tersebut. Pada kasus ini, setiap elemen queue terdiri dari dua komponen, yaitu prio dan info.

[prioqueuechar.h](#)

```
type infotype : <
    prio : integer[1..100], { prioritas dengan nilai 1..100 (1 adalah prioritas tertinggi) }
    info : char { elemen karakter }
>
```

Setiap elemen queue terurut menurun berdasarkan besaran prio sehingga elemen dengan prio yang lebih besar diantrikan di depan elemen yang memiliki prio lebih rendah. Gunakan ADT Queue yang telah Anda selesaikan sebagai tugas pra-praktikum dan adaptasikan/modifikasi menjadi ADT PrioQueueChar.

Diberikan header ADT PrioQueueChar dalam file prioqueuechar.h. Buatlah implementasi ADT PrioQueueChar dengan nama **prioqueuechar.c** yang diadaptasi dari ADT Queue pada tugas pra-praktikum.

Implementasikan semua fungsi dalam ADT PrioQueueChar. Terdapat beberapa petunjuk perubahan:

1. Elemen queue sesuai dengan informasi di atas
2. Prosedur Enqueue

Prosedur Enqueue atau penambahan elemen dilakukan sesuai dengan prioritasnya. Elemen baru akan ditambahkan di depan seluruh elemen queue yang memiliki prioritas lebih rendah dan di belakang seluruh elemen queue yang memiliki prioritas lebih tinggi. Jika terdapat elemen queue dengan prioritas yang sama, maka elemen baru akan ditambahkan sebagai elemen terakhir pada prioritas tersebut.

Contoh:

- Isi PQ saat ini : [<1,'a'>, <2,'h'>, <2,'b'>, <5,'r'>]
- Enqueue(PQ, <2,'d'>). PQ setelah penambahan: [<1,'a'>, <2,'h'>, <2,'b'>, <2,'d'>, <5,'r'>]
 - Enqueue(PQ, <6,'f'>). PQ setelah penambahan: [<1,'a'>, <2,'h'>, <2,'b'>, <2,'d'>, <5,'r'>, <6,'f'>]

3. Prosedur displayQueue/printPrioQueueChar

Prosedur ini berfungsi untuk menampilkan seluruh elemen queue. Setiap elemen ditampilkan dalam baris yang berbeda. Setiap elemen menampilkan nilai prio dan info yang dipisahkan dalam sebuah spasi. Tambahkan tanda # dan endlime pada baris terakhir. Apabila Queue kosong, pada layar hanya tercetak tanda #.

Contoh:

Isi PrioQueue (PQ)	printPrioQueueChar(PQ)
[<1,'a'>, <2,'h'>, <2,'b'>, <5,'r'>]	1 a
	2 h
	2 b
	5 r
	#
[]	#

C

 [prioqueuechar.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.42 MB
2	10	Accepted	0.00 sec, 1.47 MB
3	10	Accepted	0.00 sec, 1.52 MB
4	10	Accepted	0.00 sec, 1.51 MB
5	10	Accepted	0.00 sec, 1.47 MB
6	10	Accepted	0.00 sec, 1.49 MB
7	10	Accepted	0.00 sec, 1.54 MB
8	10	Accepted	0.00 sec, 1.46 MB
9	10	Accepted	0.00 sec, 1.52 MB
10	10	Accepted	0.00 sec, 1.47 MB

Question **3**

Not answered

Marked out of 40.00

Time limit	1 s
Memory limit	64 MB

Tuan Mike memberikan gim kepada Tuan Man. Tuan Man yang sangat ingin menjadi seorang kampiun dalam setiap gim menerima tantangan dari Tuan Mike. Tuan Mike memberikan sekumpulan bilangan positif sebanyak N yang berisi Ni bilangan bulat positif. Untuk setiap Ni, Tuan Man hanya bisa melakukan dua operasi ini.

- 1. Mengurangi nilai Ni sebanyak 1
- 2. Jika terdapat dua bilangan bulat positif a dan b yang memenuhi $a * b = Ni$, Ni dapat diganti menjadi $Ni = \max(a, b)$

Misalkan Ri adalah jumlah langkah minimum yang diperlukan untuk mengurangi nilai Ni menjadi 0. Bantulah Tuan Man untuk menentukan nilai dari hasil perkalian Ri. Anda wajib menggunakan **ADT Queue** untuk menyelesaikan soal ini.

Kumpulkan dengan nama file **mgim.c**

Format masukan

N
Ni bilangan bulat positif sebanyak N

Format keluaran

Hasil perkalian Ri

Contoh masukan 1

2
3
4

Contoh keluaran 1

9

Penjelasan 1

Untuk kasus N1 = 3, hanya terdapat 1 langkah yang mungkin, yaitu mengurangi nilai dari 3 -> 2 -> 1 -> 0 sehingga nilai R1 = 3.
Untuk kasus N2 = 4, terdapat 2 langkah yang mungkin, yaitu 4 -> 2 -> 1 -> 0 atau 4 -> 3 -> 2 -> 1 -> 0. Langkah pertama lebih minimum, sehingga R2 = 3.
Keluarkan R1 * R2 yaitu 3 * 3 sehingga bernilai 9.

Contoh masukan 2

2
6
5

Contoh keluaran 2

16

Penjelasan 2

Untuk kasus N1 = 6, langkah yang paling minimum adalah 6 -> 3 -> 2 -> 1 -> 0 sehingga R1 = 4.
Untuk kasus N2 = 5, langkah yang paling minimum adalah 5 -> 4 -> 2 -> 1 -> 0 sehingga R2 = 4.
Keluarkan R1 * R2 yaitu 4 * 4 = 16

C

Question **4**

Not answered

Marked out of 20.00

Time limit	1 s
Memory limit	64 MB

Soal ini bersifat bonus. Kerjakan jika soal lain dirasa sudah cukup.

Tuan Mike memilih sebuah daftar berisi n bilangan bulat yang unik dan sebuah bilangan k . Tuan Mike ingin agar k bilangan pertama dari daftar di reverse. Bantulah Tuan Mike dalam membuat program tersebut! Anda wajib menggunakan **ADT Queue** untuk menyelesaikan soal ini. **Anda hanya diperkenankan untuk membuat 1 Queue saja.** Gunakan **displayQueue** untuk menampilkan Queue hasil reverse.

Kumpulkan dengan nama file **mreverse.c**

Note:

Jika anda mendapatkan wrong answer atau error lain pada test case 8, 9, dan 10, berarti implementasi Queue anda masih kurang efektif, karena ketiga test case tersebut menguji kasus untuk $CAPACITY - 2 \leq n \leq CAPACITY$ dengan $CAPACITY = 100$.

Format masukan

n
 k
 n bilangan bulat yang unik

Format keluaran

Bilangan dari Queue yang direverse sesuai aturan

Contoh masukan 1

10
5
10 20 30 40 50 60 70 80 90 100

Contoh keluaran 1

[50,40,30,20,10,60,70,80,90,100]

Penjelasan 1

5 bilangan pertama dari masukan adalah 10, 20, 30, 40, dan 50. Bilangan akan dilakukan reverse sehingga menjadi 50, 40, 30, 20, 10. Sisa dari daftar bilangan dibiarkan apa adanya.

C

◀ ADT Queue - Praktikum

Jump to...

queue.h ▶