
Started on Thursday, 7 October 2021, 6:17 PM

State Finished

Completed on Thursday, 7 October 2021, 8:11 PM

Time taken 1 hour 53 mins

Marks 120.00/130.00

Grade **92.31** out of 100.00

Question 1

Correct

Mark 80.00 out of 80.00

Time limit

1 s

Memory limit

64 MB

Ambillah ADT Queue yang direpresentasikan menggunakan array eksplisit dan alokasi memori statik yang telah Anda kerjakan sebagai tugas pra-praktikum. Kapasitas dari queue sesuai header (100).

Buatlah sebuah program utama yang menggunakan ADT Queue tersebut di atas untuk mensimulasikan antrian di sebuah loket stasiun kereta (**file: antrianKereta.c**). Elemen queue menyatakan lama waktu pengunjung loket mengantri di loket tersebut (dalam menit). Program harus melakukan hal sebagai berikut:

1. Program dimulai dengan loop, pengguna bisa memilih untuk menambah pengunjung dalam antrian atau menghapus pengunjung (menghapus pengunjung berarti pengunjung tsb dilayani). Pengguna memiliki opsi sebagai berikut:
 - o Jika memilih menambah antrian, masukkan kode operasi = 1 diikuti panjang waktu mengantri (elemen queue). Contoh: 1 5 (berarti menambah elemen queue dengan waktu antrian = 5 menit).
 - o Untuk menghapus elemen antrian, masukkan kode operasi = 2.
Penambahan dan penghapusan harus mengikuti ketentuan operasi enqueue dan dequeue. Perhatikan untuk kasus queue kosong dan queue penuh. Pada operasi enqueue, jika queue penuh, tidak terjadi penambahan elemen dan berikan pesan "Queue penuh". Pada operasi dequeue, jika queue kosong, tidak terjadi penghapusan elemen dan berikan pesan "Queue kosong".
 - o Untuk menghentikan proses, masukkan kode operasi = 0.
2. Di akhir program, ditampilkan statistik queue, yaitu panjangnya antrian (berapa banyak orang yang sedang mengantri di akhir program) dan waktu rata-rata orang mengantri (dalam menit, tulis dengan 2 digit di belakang koma). Jika panjang antrian kosong, tuliskan pesan "Tidak bisa dihitung"

Buatlah fungsi sebagai berikut ke dalam program utama Anda:

```
float avgElmt (Queue Q);
/* Menghasilkan rata-rata elemen dalam queue Q yang tidak kosong */
```

Berikut adalah contoh interaksi input/output program (yang digaris bawah dan dicetak tebal adalah input pengguna):

Input/Output	Keterangan
<u>1</u> <u>5</u> <u>1</u> <u>3</u> <u>0</u> 2 4.00	Panjangnya antrian = 2 Waktu rata-rata mengantri = (5+3)/2
<u>1</u> <u>5</u> <u>1</u> <u>3</u> <u>2</u> <u>0</u> 1 3.00	Panjang antrian = 1 Waktu rata-rata mengantri = (3/1) = 3.00
<u>1</u> <u>5</u> <u>1</u> <u>3</u> <u>2</u> <u>2</u> <u>2</u> Queue kosong <u>0</u> 0 Tidak bisa dihitung	Panjang antrian = 0 Waktu rata-rata mengantri tidak bisa dihitung karena queue kosong
... <u>1</u> <u>10</u> <u>1</u> <u>10</u> <u>1</u> <u>10</u> <u>1</u> <u>10</u> Queue penuh <u>1</u> <u>9</u> Queue penuh <u>2</u> <u>2</u> <u>1</u> <u>5</u> ...	(Kasus jika Queue penuh)

Kumpulkan file **antrianKereta.c**

C

 [antrianKereta.c](#)

Score: 80

Blackbox

Score: 80

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.63 MB
2	10	Accepted	0.00 sec, 1.68 MB
3	10	Accepted	0.00 sec, 1.63 MB
4	10	Accepted	0.00 sec, 1.70 MB
5	10	Accepted	0.00 sec, 1.55 MB
6	10	Accepted	0.00 sec, 1.66 MB
7	10	Accepted	0.00 sec, 1.67 MB
8	10	Accepted	0.00 sec, 1.60 MB

Time limit	1 s
Memory limit	64 MB

Tuan Ban adalah dosen Daspro yang sadis. Karena Tuan Ban merupakan dosen yang dihormati oleh semua orang di itebe, semua peraturan dari Tuan Ban akan dituruti, termasuk peraturan antrian kantin. Pada saat bel makan siang berbunyi, semua mahasiswa akan serta merta pergi ke kantin untuk makan siang. Pada kantin itebe, terdapat peraturan khusus yakni urutan antrian pada kantin ditentukan berdasarkan nilai ujian Daspro terakhir mahasiswa. mahasiswa yang memiliki nilai Daspro lebih besar boleh menyerobot antrian. Untuk setiap mahasiswa yang mengantri, Ibu kantin akan mempersiapkan makanan satu per satu untuk mahasiswa terdepan antrian. Untuk setiap mahasiswa, diketahui nilai Daspro, waktu yang dibutuhkan ibu kantin untuk mempersiapkan makanannya serta waktu kedatangan mahasiswa di kantin. Karena hanya ada 1 ibu kantin yang mempersiapkan makanan, hanya terdapat 1 antrian pada kantin itebe.

Lihatlah tabel berikut ini ini

Waktu kedatangan	Id mahasiswa	Nama mahasiswa	Nilai Daspro	Waktu persiapan makanan (durasi layanan)
1	1	Riki	80	5
3	2	Somad	90	3
4	3	Maryo	99	2

Urutan kedatangan adalah Riki (id=1), Somad (id=2) kemudian Maryo (id=3).

- Saat waktu = 1, Riki datang. Karena kosong, Riki langsung dilayani dan dibutuhkan waktu persiapan makanan untuk Riki sebesar 5 satuan waktu (jadi akan selesai pada waktu = 5).
- Saat waktu = 3, Somad datang. Karena Riki belum selesai dilayani, Somad masuk antrian.
- Saat waktu = 4, Maryo datang. Karena Riki belum selesai dilayani, Maryo masuk ke antrian. Karena nilai Daspro Maryo lebih besar dari Somad, maka Maryo "menyerobot" antrian menjadi di urutan pertama, sedangkan Somad di belakangnya.
- Saat waktu = 5, Riki selesai dilayani.
- Saat waktu = 6, Maryo yang berada di urutan terdepan, dilayani sebesar 2 satuan waktu (jadi akan selesai pada waktu = 7).
- Saat waktu = 7, Maryo selesai dilayani.
- Saat waktu = 8, Somad baru dilayani.
- Saat waktu = 10, Somad selesai dilayani.

Somad membutuhkan total **waktu tunggu hingga mulai dilayani** di kantin sebesar 5 satuan waktu semenjak kedatangannya sampai makanannya mulai dipersiapkan, sedangkan Maryo membutuhkan waktu tunggu sebesar 2 satuan waktu dan Riki hanya membutuhkan waktu tunggu sebesar 0.

Ilustrasi antrian per waktu:

Waktu	1	2	3	4	5	6	7	8	9	10
Dilayani	Riki	Riki	Riki	Riki	Riki	Maryo	Maryo	Somad	Somad	Somad
Dalam antrian	[]	[Somad]	[Somad]	[Maryo, Somad]	[Maryo, Somad]	[Somad]	[Somad]	[]	[]	[]

Sebuah program digunakan untuk mensimulasikan persoalan di atas. Implementasikan dan gunakan **ADT Priority Queue** yang direpresentasikan dengan array eksplisit dengan alokasi statik (diberikan dalam file [priorityqueue.h](#)). Berikut adalah struktur dari EType:

```
typedef struct {
    int id;           /* id mahasiswa */
    int tArrival;     /* waktu kedatangan */
    int score;        /* skor Daspro */
    int dService;     /* durasi/lama persiapan makanan untuk mahasiswa */
} EType;
```

Format Input:

- Baris pertama berisi sebuah bilangan N yang menandakan banyaknya mahasiswa.
- Baris kedua sampai N+1: terdiri atas 4 buah bilangan, misalnya T, I, M, D yang masing-masing merepresentasikan: T: waktu kedatangan ke kantin, I: id mahasiswa, M: nilai Daspro, dan D: durasi layanan/lama persiapan makanan untuk mahasiswa I.

Asumsi dan Batasan:

- Nilai Daspro (M) tidak dijamin berbeda. Bila 2 mahasiswa atau lebih memiliki nilai Daspro yang sama, maka diurutkan berdasarkan waktu kedatangan (T).
- Waktu kedatangan (T) dijamin unik.
- Input terurut membesar berdasarkan waktu kedatangan (T).
- N, T, I, M, D adalah bilangan bulat. $N \geq 0$, $T > 0$, $I > 0$, $0 \leq M \leq 100$, $0 < D \leq 20$.

Format Output:

Jika $N > 0$, output terdiri atas N buah baris, tiap baris berisi $\langle t, i, W \rangle$ dengan t adalah waktu mulai dilayani, i merupakan id mahasiswa, dan W merupakan total waktu tunggu di kantin untuk mahasiswa dengan id i. Urutan penulisan adalah sesuai dengan urutan mahasiswa dilayani.
Jika $N = 0$, tuliskan "Tidak ada layanan"

Contoh Input-Output

Input	Output
-------	--------

3 1 1 80 5 3 2 90 3 4 3 99 2	1 1 0 6 3 2 8 2 5
2 1 33 90 1 2 35 99 2	1 33 0 2 35 0
3 1 1 80 11 9 2 92 10 10 3 90 2	1 1 0 12 2 3 22 3 12
5 1 11 70 5 8 22 80 5 9 33 90 2 10 44 90 1 20 55 99 2	1 11 0 8 22 0 13 33 4 15 44 5 20 55 0
0	Tidak ada layanan

Yang harus di-submit: file **prioqueue.c** dan **mprioqueue.c** (zip dalam file **antrian.zip** tanpa dimasukkan ke dalam folder)

C

 [antrian.zip](#)

Score: 40

Blackbox

Score: 40

Verdict: Time limit exceeded

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.50 MB
2	10	Accepted	0.00 sec, 1.67 MB
3	10	Accepted	0.00 sec, 1.61 MB
4	0	Time limit exceeded	1.00 sec, 1.54 MB
5	10	Accepted	0.00 sec, 1.50 MB

[◀ prioqueue.h](#)

Jump to...

[stack.h ▶](#)