

Started on	Sunday, 17 November 2024, 7:05 PM
State	Finished
Completed on	Sunday, 17 November 2024, 8:38 PM
Time taken	1 hour 32 mins
Grade	400.00 out of 400.00 (100%)

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Download file header ADT Priority Queue dalam [prioqueue.h](#) dan buatlah file implementasinya. Kumpulkan hanya file **prioqueue.c**.

Catatan:

- Priority Queue adalah Queue yang terurut mengecil. Contoh: Queue = [1, 4, 2, 3] -> Priority Queue = [4, 3, 2, 1].
- Dapat menggunakan ADT Queue yang telah dibuat sebelumnya sebagai referensi.
- Buatlah driver sendiri untuk mengetes setiap fungsi/prosedur yang ada.

C

 [prioqueue.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	4	Accepted	0.00 sec, 1.64 MB
2	4	Accepted	0.00 sec, 1.52 MB
3	4	Accepted	0.00 sec, 1.64 MB
4	4	Accepted	0.00 sec, 1.60 MB
5	4	Accepted	0.00 sec, 1.66 MB
6	4	Accepted	0.00 sec, 1.66 MB
7	4	Accepted	0.00 sec, 1.64 MB
8	4	Accepted	0.00 sec, 1.63 MB
9	4	Accepted	0.01 sec, 1.61 MB
10	4	Accepted	0.00 sec, 1.50 MB
11	4	Accepted	0.00 sec, 1.64 MB
12	4	Accepted	0.00 sec, 1.49 MB
13	4	Accepted	0.00 sec, 1.57 MB
14	4	Accepted	0.00 sec, 1.60 MB
15	4	Accepted	0.00 sec, 1.60 MB
16	4	Accepted	0.00 sec, 1.60 MB
17	4	Accepted	0.00 sec, 1.62 MB
18	4	Accepted	0.00 sec, 1.64 MB
19	4	Accepted	0.00 sec, 1.50 MB
20	4	Accepted	0.00 sec, 1.55 MB
21	4	Accepted	0.00 sec, 1.66 MB
22	4	Accepted	0.00 sec, 1.64 MB
23	4	Accepted	0.00 sec, 1.59 MB

No	Score	Verdict	Description
24	8	Accepted	0.00 sec, 1.65 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Message Queue

Deskripsi

Message Queue adalah sebuah teknologi antrian pesan yang dapat mengakomodasi komunikasi antar servis di suatu perangkat lunak.

Pada soal ini, kalian diminta untuk membuat sebuah program simulasi Message Queue yang hanya berisi angka-angka. Program akan menerima 4 tipe query sebagai berikut.

- **1 x.** Apabila queue penuh, program hanya menampilkan pesan "Queue Penuh" diakhiri dengan newline. Apabila queue tidak penuh, program memasukkan x ke dalam queue dan menampilkan pesan "Enqueue x Berhasil" diakhiri dengan newline.
- **2.** Apabila queue kosong, program hanya menampilkan pesan "Queue Kosong" diakhiri dengan newline. Apabila queue tidak kosong, program menghapus elemen paling depan dan menampilkan pesan "Dequeue x Berhasil" diakhiri dengan newline dimana x adalah elemen terdepan pada Queue.
- **3.** Apabila queue kosong, program hanya menampilkan pesan "Queue Kosong" diakhiri dengan newline. Apabila queue tidak kosong, program membaca elemen paling depan dan menampilkan pesan "Elemen Awal Queue adalah x" diakhiri dengan newline dimana x adalah elemen terdepan pada Queue.
- **4.** Program menampilkan pesan "Ukuran Queue saat ini adalah x" diakhiri dengan newline dimana x adalah ukuran atau banyak elemen pada Queue.

Setelah memproses seluruh query, program akan melakukan **displayQueue** sesuai dengan ADT untuk menampilkan kondisi akhir Queue. Buatlah program bernama [messagequeue.c](#) yang dapat memproses 4 tipe query tersebut **menggunakan ADT Queue** yang telah kalian buat.

Format Masukan

- Baris pertama berisi sebuah bilangan bulat N ($1 \leq N \leq 1000$).
- N baris selanjutnya berisi *query* yang bisa berbentuk 1 x, 2, 3, dan 4.

Format Keluaran

N baris pertama keluaran sesuai dengan penjelasan masing-masing *query*.

Contoh

No	Masukan	Keluaran
1.	5 2 3 4 1 10 4	Queue Kosong Queue Kosong Ukuran Queue saat ini adalah 0 Enqueue 10 Berhasil Ukuran Queue saat ini adalah 1 [10]
2.	1000 1 5 ... (Enqueue sebanyak 98 kali) 1 100 1 10 2 1 10 Enqueue 100 Berhasil Queue Penuh Dequeue 5 Berhasil Enqueue 10 Berhasil ...

Penjelasan

Pada contoh kedua, program telah melakukan enqueue 100 kali sehingga queue penuh dan enqueue 10 pertama gagal. Lalu, program melakukan deque 1 kali dan enqueue 10 selanjutnya berhasil.

Catatan

- Setiap keluaran diakhiri dengan newline (\n)
- Download dan gunakan file [boolean.h](#), [queue.h](#) & [queue.c](#) yang telah dibuat sebelumnya
- Download dan kumpulkan jawaban dalam file bernama [messagequeue.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.60 MB
2	10	Accepted	0.00 sec, 1.58 MB
3	10	Accepted	0.00 sec, 1.50 MB
4	10	Accepted	0.00 sec, 1.55 MB
5	10	Accepted	0.00 sec, 1.50 MB
6	10	Accepted	0.00 sec, 1.62 MB
7	10	Accepted	0.00 sec, 1.71 MB
8	10	Accepted	0.00 sec, 1.64 MB
9	10	Accepted	0.00 sec, 1.62 MB
10	10	Accepted	0.00 sec, 1.56 MB

Time limit	1 s
Memory limit	64 MB

BFS

Deskripsi

Bayangkan terdapat sebuah **Grid** berukuran $N \times N$ yang kotak-kotaknya mungkin memiliki jebakan ('#'). Tidak diperbolehkan untuk berpindah ke kotak yang memiliki jebakan.

Tugas Anda adalah mencari jalur terpendek dari titik awal ('A') ke titik akhir ('B'). Anda bisa bergerak ke kanan, kiri, atas, dan bawah.

Format Masukan

- Baris pertama berisi sebuah bilangan bulat N ($2 \leq N \leq 100$).
- Baris kedua berisi $N \times N$ kotak. Tiap kotak dapat berupa karakter '.' (aman), '#' (jebakan), 'A' (titik awal), dan 'B' (titik akhir).
- Dijamin bahwa pasti ada tepat satu titik awal dan titik akhir.

Format Keluaran

Pertama keluarkan "Ya", jika terdapat jalur dari titik awal ke titik akhir, selain itu keluarkan "Tidak".

Jika terdapat jalur, keluarkan juga panjang jalur terpendek untuk mencapai titik akhir.

Contoh

No	Masukan	Keluaran
1.	3 A.# #.. B..	Ya 4
2.	2 A# #B	Tidak

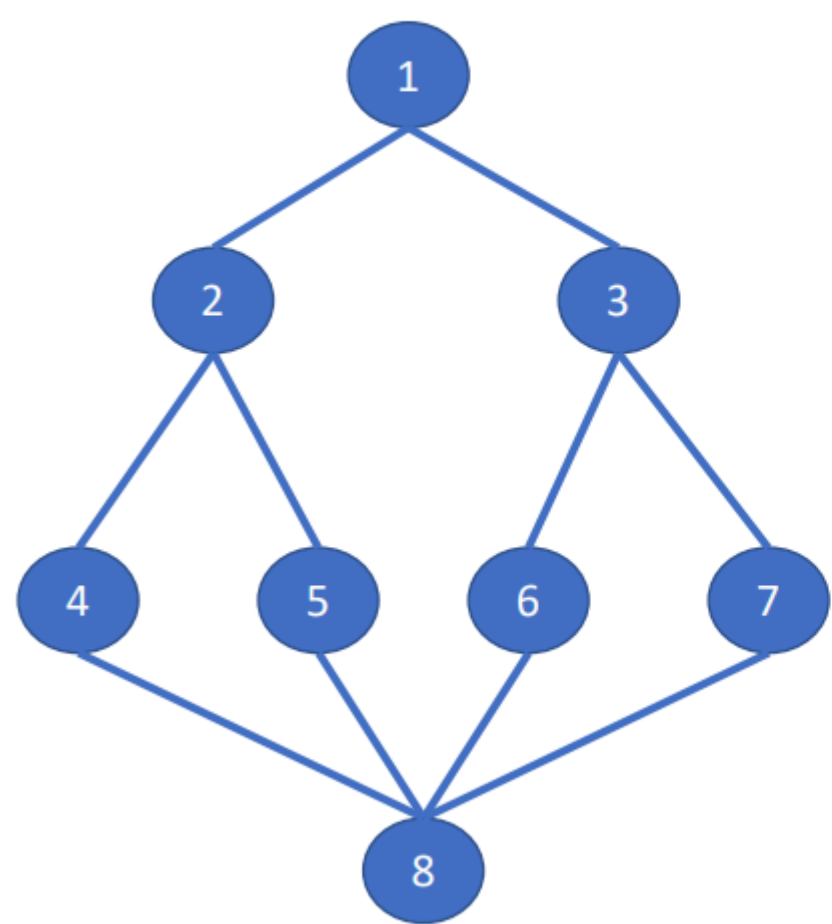
Penjelasan

Untuk contoh 1, jalur terpendek dari titik awal ke titik akhir ialah Kanan-Bawah-Bawah-Kiri sehingga panjang jalurnya menjadi 4.

Catatan

- Keluaran diakhiri dengan newline (\n)
- Download dan gunakan file [boolean.h](#), [queue.h](#) (terdapat perubahan pada CAPACITY & EType) & [queue.c](#) yang telah dibuat sebelumnya
- Download dan kumpulkan jawaban dalam file bernama [bfs.c](#)

BFS: Ilustrasi



Iterasi	V	Q	dikunjungi			
			1	2	3	4
Inisialisasi	1	{1}	T	F	F	F
Iterasi 1	1	{2,3}	T	T	T	F
Iterasi 2	2	{3,4,5}	T	T	T	T
Iterasi 3	3	{4,5,6,7}	T	T	T	T
Iterasi 4	4	{5,6,7,8}	T	T	T	T
Iterasi 5	5	{6,7,8}	T	T	T	T
Iterasi 6	6	{7,8}	T	T	T	T
Iterasi 7	7	{8}	T	T	T	T
Iterasi 8	8	{}	T	T	T	T

Urutan simpul2 yang dikunjungi: 1, 2,

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, 1.75 MB
3	10	Accepted	0.00 sec, 1.75 MB
4	10	Accepted	0.00 sec, 1.59 MB
5	10	Accepted	0.00 sec, 1.59 MB
6	10	Accepted	0.00 sec, 1.87 MB
7	10	Accepted	0.08 sec, 1.89 MB
8	10	Accepted	0.00 sec, 1.82 MB
9	10	Accepted	0.00 sec, 1.88 MB
10	10	Accepted	0.00 sec, 1.82 MB

Question **4**

Correct

Mark 100.00 out of 100.00

Time limit	5 s
Memory limit	64 MB

Maxxing 2

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

Deskripsi

Tuan Aqua sedang bermain game gacha yang bergenre RPG. Tuan Aqua saat ini memiliki N buah karakter dan N buah senjata yang dapat dipasangkan secara bebas. Setiap karakter memiliki **kekuatan A_i** dan setiap senjata memiliki **kekuatan B_i** . Apabila seorang karakter dipasangkan dengan sebuah senjata, **kekuatan tim Tuan Aqua** bertambah sebesar **$A_i \times B_i$** .

Karena Tuan Aqua tidak menyukai matematika, dia meminta bantuanmu untuk membantu memasangkan setiap karakter dengan senjatanya. Buatlah program bernama [maxxing.c](#) yang dapat menghasilkan **kekuatan maksimum tim Tuan Aqua** menggunakan ADT Priority Queue yang telah kalian buat.

Format Masukan

- Baris pertama berisi sebuah bilangan bulat N ($1 \leq N \leq 10000$).
- Baris selanjutnya berisi N bilangan yang masing-masing merepresentasikan A_i ($0 \leq A_i \leq 1000000$) yaitu kekuatan karakter ke- i .
- Baris selanjutnya berisi N bilangan yang masing-masing merepresentasikan B_i ($0 \leq B_i \leq 1000000$) yaitu kekuatan senjata ke- i .

Format Keluaran

Sebuah bilangan yang bernilai kekuatan maksimum tim Tuan Aqua.

Contoh

No	Masukan	Keluaran
1.	3 2 4 6 5 3 1	44
2.	3 1 0 0 0 0 1	1

Penjelasan

Pada contoh pertama, 44 dicapai dengan pasangan berikut:

$$6 \times 5 + 4 \times 3 + 2 \times 1 = 30 + 12 + 2 = 44$$

Pada contoh kedua, pasangan yang memiliki nilai lebih dari 0 hanya 1×1 maka keluarannya bernilai 1.

Catatan

- Keluaran diakhiri dengan newline (`\n`)
- Gunakan tipe data **long long** untuk jawaban untuk mencegah integer overflow. Gunakan **%lld** untuk melakukan output data long long.
- Download dan gunakan file [boolean.h](#), [prioqueue.h](#) (Ada perubahan nilai CAPACITY) & **prioqueue.c** yang telah dibuat sebelumnya
- Download dan kumpulkan jawaban dalam file bernama [maxxing.c](#)



[maxxing.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.03 MB
2	10	Accepted	0.00 sec, 2.11 MB
3	10	Accepted	0.33 sec, 1.96 MB
4	10	Accepted	0.75 sec, 2.16 MB
5	10	Accepted	0.74 sec, 2.19 MB
6	10	Accepted	0.76 sec, 2.11 MB
7	10	Accepted	0.76 sec, 2.16 MB
8	10	Accepted	0.74 sec, 2.11 MB
9	10	Accepted	0.76 sec, 2.19 MB
10	10	Accepted	0.74 sec, 1.96 MB

[◀ Queue - Praktikum](#)

Jump to...

[Linked List - Pra Praktikum ▶](#)