

Started on	Wednesday, 4 June 2025, 10:08 PM
State	Finished
Completed on	Wednesday, 4 June 2025, 11:39 PM
Time taken	1 hour 31 mins
Marks	300.00/300.00
Grade	10.00 out of 10.00 (100%)

Question 1

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: **graph.c**

Pada suatu hari, Gro memberikan tantangan untuk membuat sistem representasi graf. Kebin diminta untuk membuat program sederhana yang dapat memodelkan relasi pertemanan antar karakter di sebuah game. Setiap karakter akan direpresentasikan sebagai simpul (vertex), dan jika dua karakter saling mengenal, maka akan ada sisi (edge) di antara keduanya. Karena hubungan pertemanan bersifat timbal balik, graf yang digunakan harus **tak berarah**. Gro sudah memberikan file yang berisi deklarasi tipe data dan fungsi-fungsi yang harus diimplementasikan. Kini Kebin bingung dan membutuhkan bantuanmu untuk menyelesaikan tantangan ini.

Bantulah Kebin dengan melengkapi [graph.h](#). Kumpulkan hanya **graph.c**.

Gunakan [main.c](#) untuk menguji program yang dibuat.

C

 [graph.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 1.60 MB
2	20	Accepted	0.00 sec, 1.75 MB
3	20	Accepted	0.00 sec, 1.74 MB
4	20	Accepted	0.00 sec, 1.74 MB
5	20	Accepted	0.00 sec, 1.54 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: graphMultiList.c

Nimons diberi tugas oleh Gro untuk mempelajari graf. Awalnya Gro menugasi Nimons untuk mempelajari graf dari konsepnya terlebih dahulu. Namun, Dr. Neroifa menganggap hal tersebut terlalu lama, jadi beliau memaksa Nimons untuk belajar graf dengan representasi **Multi List**.

Untuk itu, kamu akan mengimplementasikan **ADT Directed Graph** dengan representasi **Multilist** pada file [graphMultiList.h](#)

Anda hanya diminta mengumpulkan `graphMultiList.c` nya saja. Selain itu, tersedia pula file [boolean.h](#) yang dapat digunakan bila diperlukan dalam implementasi Anda.

Batasan

- $1 \leq \text{ID Nimons} \leq 1000$
- $1 \leq \text{Banyaknya Node} \leq 100$
- $0 \leq \text{Banyaknya edge} \leq 500$
- Tiap ID Nimons unik.
- Setiap hubungan direct ($A \rightarrow B$) hanya dicatat **sekali**.
- Tidak ada self-direct, Nimons tidak boleh menunjuk dirinya sendiri.

Gunakan [main.c](#) untuk menguji program yang Anda buat.

Keluaran yang diharapkan dari main.c tersebut:

Graph saat ini:

```
1 -> 2 -> 3
2 -> 3
3 -> 4 -> 5
4 -> 3 -> 2
5 -> 2 -> 3 -> 4
10
```

Graph saat ini:

```
1 -> 2 -> 3
2 -> 3
3 -> 4
4 -> 3 -> 2
10
```

Graph saat ini:

```
1 -> 2 -> 3
2 -> 3
3 -> 4
4 -> 3 -> 2
10
```

Graph saat ini:

```
1 -> 2
2
4 -> 2
10
```

Graph saat ini:

```
1 -> 2
2
4 -> 2
10
```

NOTE:

Pastikan setiap output diakhiri oleh **endline** ("**\n**")!

C

 [graphMultiList.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.46 MB
2	10	Accepted	0.00 sec, 1.66 MB
3	10	Accepted	0.00 sec, 1.64 MB
4	10	Accepted	0.00 sec, 1.66 MB
5	10	Accepted	0.00 sec, 1.64 MB
6	10	Accepted	0.00 sec, 1.49 MB
7	10	Accepted	0.00 sec, 1.60 MB
8	10	Accepted	0.00 sec, 1.66 MB
9	10	Accepted	0.00 sec, 1.50 MB
10	10	Accepted	0.00 sec, 1.66 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Nama File: graph_adjlist.c

Di markas Gro, para Nimons sedang menghadapi masalah baru. Mereka mendapatkan tugas dari Luiy, agen rahasia dari AVL (Anti-Villain League), untuk memantau jaringan distribusi pisang ilegal yang tersebar di berbagai kota. Gro yang kini bekerja sama dengan AVL membutuhkan sistem pelacakan yang efisien untuk mengidentifikasi rute penyebaran pisang dan hubungan antar kota yang mencurigakan.

Toto, seorang Nimon yang dikenal jenius dalam matematika graf, menyarankan untuk memodelkan jaringan kota-kota tersebut dalam bentuk **Graph**, di mana setiap kota adalah *node* dan hubungan transportasi antar kota adalah *edge*. Untuk efisiensi penyimpanan dan kemudahan penambahan rute baru, Toto mengusulkan menggunakan representasi **Adjacency List**.

Dave, Pop, Kebin, dan Stewart pun mulai bekerja. Mereka harus membuat ADT Graph menggunakan adjacency list dalam bahasa C. Graph ini bersifat **tidak berarah (undirected)** karena jalur transportasi pisang dapat dilewati dua arah. Mereka juga harus menyediakan fungsi untuk menambahkan node (kota), menambahkan edge (rute), mencetak semua koneksi yang ada, dan mencari tahu apakah dua kota terhubung secara langsung.

Namun, mereka menghadapi beberapa tantangan: bagaimana menyimpan daftar koneksi dengan efisien, bagaimana memastikan tidak ada edge ganda (duplikat rute), dan bagaimana menghindari memory leak saat node-node dihapus.

Bantulah para Nimon untuk menyelesaikan implementasi ini dengan melengkapi file `graph_adjlist.c`. Header file yang digunakan adalah [graph_adjlist.h](#).

Catatan: Graph harus diimplementasikan menggunakan adjacency list, bukan adjacency matrix.

C

 [graph_adjlist.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.00 sec, 1.49 MB
2	20	Accepted	0.00 sec, 1.64 MB
3	20	Accepted	0.00 sec, 1.59 MB
4	20	Accepted	0.00 sec, 1.60 MB
5	20	Accepted	0.00 sec, 1.60 MB

[◀ Post Praktikum 9 - K3 & K4](#)

Jump to...

[Praktikum 10 - K3 & K4 ▶](#)