

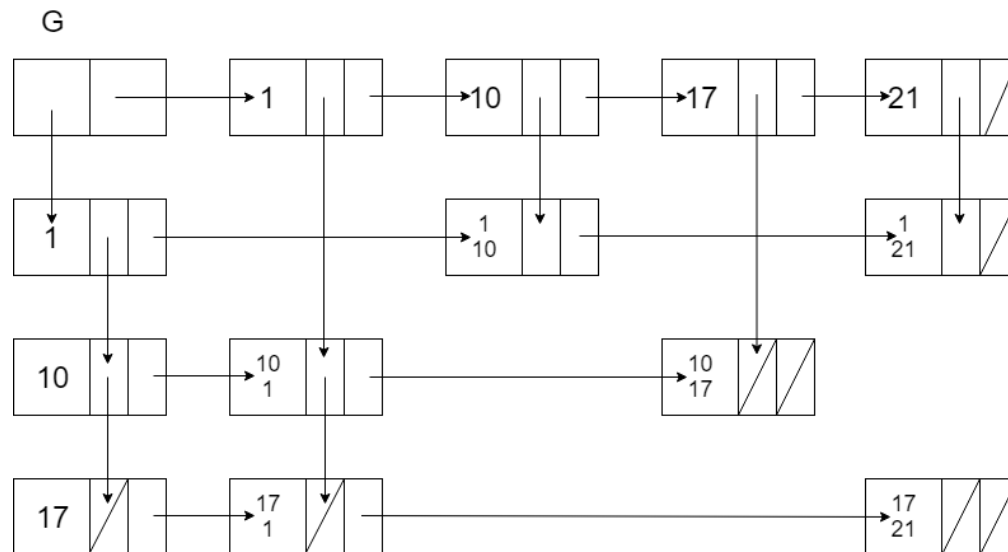
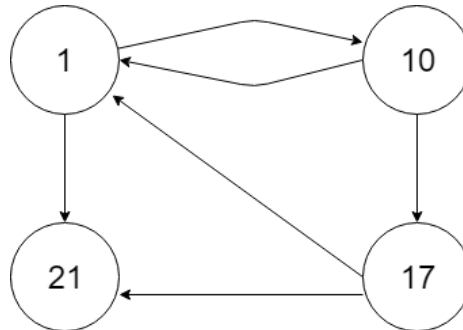
Ketentuan Tugas Pendahuluan

- Pengerjaan TP pada buku TP, yaitu buku ukuran B5 dengan sampul berwarna putih dan cover TP yang telah ditentukan.
- TP dikumpulkan **maksimal hari Senin pukul 09.00 WIFLAB** di LOKER sesuai kelas masing-masing.
- Asisten tidak bertanggung jawab terhadap TP yang tidak berada pada tempat yang ditentukan sebelumnya.
- TP ini bersifat **WAJIB, TIDAK MENGERJAKAN BERARTI TIDAK BOLEH MENGIKUTI PRAKTIKUM.**
- **Keterlambatan** pengumpulan TP maka nilai **TP dikurangi 75%.**
- **Tidak mengerjakan satu soal** TP maka nilai **TP = 0 dan Tes Awal = 0.**
- Kerjakan TP dengan jelas agar dapat dimengerti.
- Untuk soal teori **DITULIS TANGAN.**
- Untuk soal algoritma **WAJIB MENYERTAKAN SCREENSHOT CODINGAN C++ DAN HASIL OUTPUTNYA.**
- Screenshot ditempelkan pada buku TP.
- Untuk setiap soal algoritma dibawahnya diberikan comment **NAMA** dan **NIM** seperti dibawah.

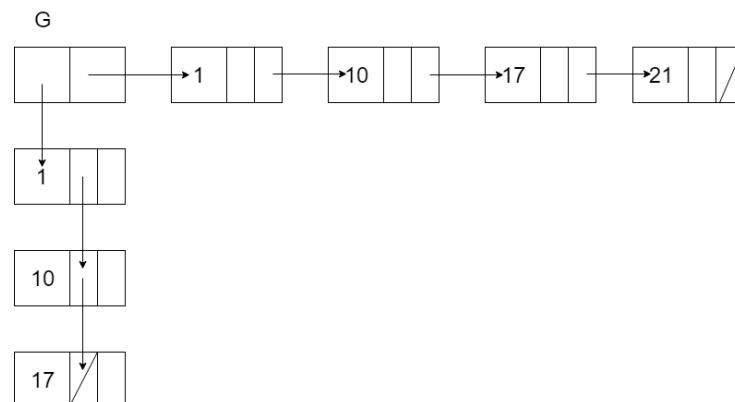
```
int example (int a, int b) {  
    /*  
    Name : Ichi Ocha  
    NIM : 1301123456  
    */  
    return a + b;  
}
```

BAGIAN 1

Di bawah ini adalah contoh graf berarah dan representasi lojiknya dengan linked list.



Untuk tugas pendahuluan ini, anda akan mendapatkan file graph.h dan graph.cpp untuk tipe data, fungsi, prosedur untuk graf berikut ini:



Tugas anda adalah mempelajari setiap fungsi dan prosedur tersebut. Kemudian, panggil fungsi createGraph dan tampilkan isi graf. Contoh tampilan :

```
Hasil Output setelah createGraph() seharusnya:  
Simpul asal : 1, 10, 17,  
Simpul tujuan : 1, 10, 17, 21,  
Contoh Output Seharusnya
```

BAGIAN 2

1. Panggil fungsi createGraph untuk membuat sebuah grup G yang berisi semua simpul asal dan tujuan.
2. Buat fungsi-fungsi di bawah ini:

function sourceVertex (numVertex: integer) → adrVertex
{Mengecek apakah vertex dengan nomor **numVertex** merupakan simpul **asal** dari sebuah edge. Jika ya, fungsi mengirimkan alamat vertex tersebut, atau null jika sebaliknya.}

function destinationVertex (numVertex: integer) → adrVertex
{Mengecek apakah vertex dengan nomor **numVertex** merupakan simpul **tujuan** dari sebuah edge. Jika ya, fungsi mengirimkan alamat vertex tersebut, atau null jika sebaliknya.}
3. Buat prosedur **addEdge** untuk menambahkan sebuah edge. Gunakan fungsi **sourceVertex** dan **destinationVertex** jika perlu. (disarankan *insertFirst*)
4. Panggil prosedur addEdge secara berulang untuk memasukkan edge:

17-21 (artinya dari **17** ke **21**)

17-1

10-17

10-1

1-21

1-10

Sehingga terbentuk graf seperti di halaman 1.