Program yang telah saya buat adalah sebuah implementasi dari struktur data antrian (queue) menggunakan pendekatan linked list. Program ini memungkinkan pengguna untuk melakukan beberapa operasi pada antrian, seperti menambahkan elemen ke antrian, menghapus elemen dari antrian, menampilkan elemen depan dan belakang antrian, serta memeriksa apakah antrian kosong.

Berikut adalah penjelasan detail tentang program tersebut:

1. Deklarasi Struktur Data:
   * **nodeType<Type>**: Struktur data ini digunakan untuk merepresentasikan simpul (node) dalam linked list. Setiap simpul memiliki dua anggota, yaitu **info** untuk menyimpan nilai elemen, dan **link** untuk menunjukkan ke simpul berikutnya.
   * **linkedQueueType<Type>**: Kelas ini mewakili antrian menggunakan linked list. Setiap antrian memiliki dua pointer, yaitu **queueFront** yang menunjukkan elemen depan antrian, dan **queueRear** yang menunjukkan elemen belakang antrian.
2. Fungsi dan Metode Utama:
   * **linkedQueueType<Type>::linkedQueueType()**: Konstruktor kelas yang menginisialisasi pointer **queueFront** dan **queueRear** menjadi **NULL**.
   * **linkedQueueType<Type>::~linkedQueueType()**: Destruktor kelas yang menghapus semua elemen dalam antrian dengan memanggil **deleteQueue()** secara berulang hingga antrian kosong.
   * **linkedQueueType<Type>::isEmptyQueue()**: Metode ini memeriksa apakah antrian kosong atau tidak. Jika **queueFront** bernilai **NULL**, maka antrian dianggap kosong.
   * **linkedQueueType<Type>::addQueue(const Type& newElement)**: Metode ini digunakan untuk menambahkan elemen baru ke dalam antrian. Sebuah simpul baru dibuat dengan nilai elemen baru, dan kemudian simpul tersebut ditambahkan di belakang antrian.
   * **linkedQueueType<Type>::deleteQueue()**: Metode ini menghapus elemen dari depan antrian. Jika antrian tidak kosong, elemen depan dihapus dan pointer **queueFront** dipindahkan ke simpul berikutnya.
   * **linkedQueueType<Type>::front()**: Metode ini mengembalikan nilai elemen yang berada di depan antrian. Jika antrian tidak kosong, nilai elemen depan dikembalikan.
   * **linkedQueueType<Type>::back()**: Metode ini mengembalikan nilai elemen yang berada di belakang antrian. Jika antrian tidak kosong, nilai elemen belakang dikembalikan.
3. Fungsi **main()**:
   * Fungsi **main()** adalah fungsi utama program yang menampilkan menu dan memungkinkan pengguna untuk memilih operasi yang ingin dilakukan pada antrian.
   * Pengguna dapat memilih opsi untuk menambahkan elemen ke antrian, menghapus elemen dari antrian, menampilkan elemen depan dan belakang antrian, memeriksa apakah antrian kosong, dan keluar dari program.
   * Setiap operasi yang dipilih akan memanggil metode terkait dari kelas **linkedQueueType** untuk melakukan tindakan yang sesuai pada antrian.