



Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)
Tahun / Semester : 2021-2022 / Genap
Modul : 6 - Linked List
Hari, Tanggal Praktikum : Rabu, 30 Maret 2022

Naskah Soal Praktikum

Pembuat Naskah: Syifa Kushirayati, Agape D'sky

Ketentuan:

1. Kerjakanlah satu dari dua soal berikut pada *template repository* yang anda peroleh ketika mengambil *assignment* di GitHub Classroom praktikum!
2. *Commit* yang dilakukan setelah sesi praktikum berakhir tidak akan dipertimbangkan dalam penilaian.
3. *Header* setiap *file* harus mengikuti format yang telah disediakan pada *file template repository*. *Header* yang tidak mengikuti format tersebut tidak akan dinilai.
4. Buku catatan laboratorium yang berisi alasan pemilihan soal, *flowchart*, dan *data flow diagram* dari solusi yang anda buat dikumpulkan ke tugas.stei.itb.ac.id paling lambat pukul 11.00 WIB satu hari kerja setelah sesi praktikum.
5. Solusi soal pertama harus dapat dikompilasi dengan perintah `make soal-01` dan menghasilkan *file executable* dengan nama `soal-01`. Demikian pula, soal kedua harus dapat dikompilasi dengan perintah `make soal-02` dan menghasilkan *file executable* dengan nama `soal-02`.
6. Bila diperlukan, sesuaikanlah isi *Makefile* yang tersedia pada *template repository* untuk memenuhi syarat kompilasi dan *file* keluaran di atas!

Soal 1

Buatlah program yang dapat melakukan pengeditan *insertion* dan atau *deletion* terhadap susunan basa nitrogen :

Misalnya,

Contoh susunan basa nitrogen: **AAATTTGGGCCC**

***Maksud urutan basa ke-0 adalah NULL, urutan basa ke-1 adalah A**

Setelah dilakukan *insertion* **AT** setelah urutan basa ke 5 : **AAATTATTGGGCCC** (lihat contoh eksekusi program #1)

Setelah dilakukan *deletion* pada basa nitrogen setelah urutan ke 0 sampai 2 : **ATTTGGGCCC** (lihat contoh eksekusi program #2)

Hal yang perlu Anda perhatikan:

1. Asumsi panjang urutan basa terbatas
2. Masukkan pilihan operasi dianggap selalu valid
3. Masukkan urutan basa dianggap selalu valid (berupa string berurutan tidak ada spasi seperti di contoh eksekusi program)
4. Masukkan urutan basa ke - n untuk operasi insert dianggap selalu valid
5. Masukkan urutan basa ke - n sampai ke - m untuk operasi delete dianggap selalu valid

Contoh Eksekusi Program (garis bawah menandakan input)

#1

```
Masukkan urutan basa yang akan diedit: AAATTTGGGCCC
Pilih operasi 1.insertion 2.deletion 3.exit: 1
Masukkan urutan basa yang akan di insert: AT
Setelah urutan basa ke- berapa akan di insert urutan basa baru: 5
Pilih operasi 1.insertion 2.deletion 3.exit: 3
Hasil edit urutan basa nitrogen: AAATTATTGGGCCC
```

#2

```
Masukkan urutan basa yang akan diedit: AAATTTGGGCCC
Pilih operasi 1.insertion 2.deletion 3.exit: 2
Urutan basa dari setelah berapa sampai berapa yang akan di delete:
0 2
Pilih operasi 1.insertion 2.deletion 3.exit: 3
Hasil edit urutan basa nitrogen: ATTTGGGCCC
```

#3

Masukkan urutan basa yang akan diedit: CCGGTTAGCGATGACGGTA

Pilih operasi 1.insertion 2.deletion 3.exit: 1

Masukkan urutan basa yang akan di insert: AAAAA

Setelah urutan basa ke- berapa akan di insert urutan basa baru: 0

Pilih operasi 1.insertion 2.deletion 3.exit: 3

Hasil edit urutan basa nitrogen: AAAAACCGGTTAGCGATGACGGT

#4

Masukkan urutan basa yang akan diedit: ATGGCCTTAAGCCTGA

Pilih operasi 1.insertion 2.deletion 3.exit: 2

Urutan basa dari setelah berapa sampai berapa yang akan di delete:

0 5

Pilih operasi 1.insertion 2.deletion 3.exit: 1

Masukkan urutan basa yang akan di insert: AATGC

Setelah urutan basa ke- berapa akan di insert urutan basa baru: 1

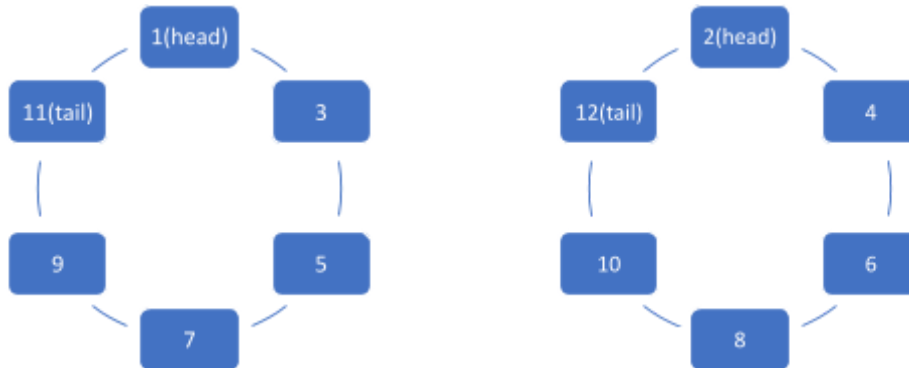
Pilih operasi 1.insertion 2.deletion 3.exit: 3

Hasil edit urutan basa nitrogen: CAATGCTTAAGCCTGA

Soal 2

Linked list bisa dibuat dalam struktur sirkular, yakni elemen terakhir dari *linked list* dihubungkan dengan elemen pertamanya (dalam notasi c, $\text{tail} \rightarrow \text{next} = \text{head}$). *Linked list* jenis ini bisa digunakan sebagai dasar dalam pemrosesan filter digital, yakni dengan konvolusi sirkular. Namun, untuk simplisitas, pada soal ini praktikan hanya akan diminta untuk melakukan peragaan operasi satu elemen konvolusi.

Misalkan, diberikan dua buah *linked list* sirkular, seperti pada ilustrasi berikut:

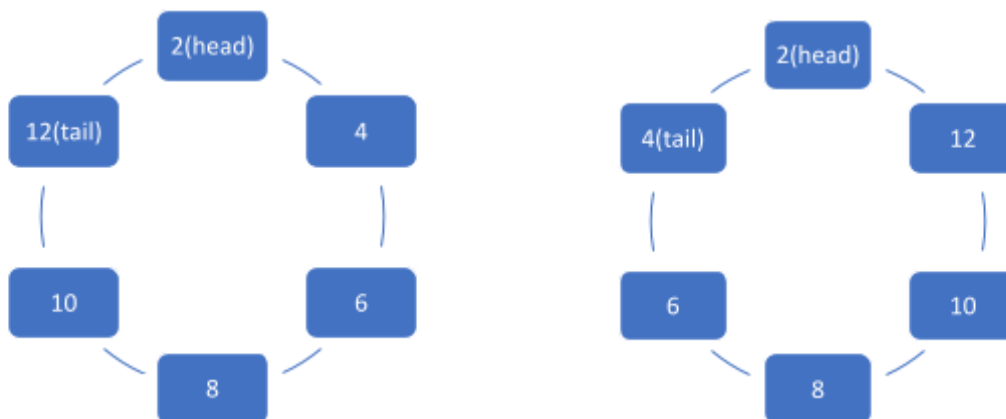


Satu elemen konvolusi bisa dihitung dengan cara korespondensi satu-satu antara *circular linked list* kiri dan kanan.

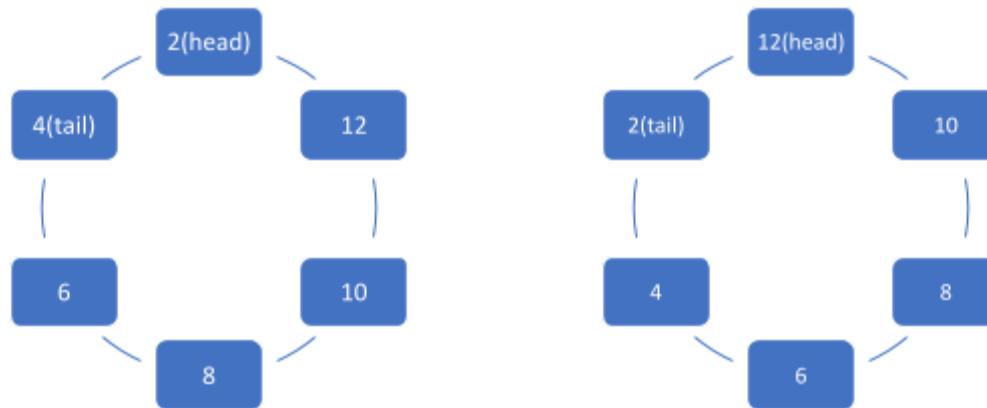
$$\text{Hasil} = 1 * 2 + 3 * 4 + 5 * 6 + 7 * 8 + 9 * 10 + 11 * 12 = 322$$

Praktikan diminta untuk membuat program yang menerima 2 buah *linked list* sirkular, lalu melakukan operasi seperti ilustrasi di atas.

Namun, ada 2 input tambahan, yakni nilai *shift* dan *flip* dari *linked list* 2. Jika dilakukan *flipping*, maka arah gerak konvolusi perlu diubah terlebih dahulu sebelum dikonvolusi.



Ilustrasi Flipping



Ilustrasi *Shifting* +1

Ketentuan tambahan:

- Input selalu dianggap selalu valid, berisi informasi nilai-nilai pada *linked list* pertama dan kedua. Kedua *linked list* memiliki jumlah elemen yang sama
- Data pada *linked list* selalu berupa integer, anggap konvolusi tidak pernah memberikan hasil *overflow*
- *Shift* bisa bernilai positif dan negatif. Jika tidak dilakukan *shifting*, maka diberikan nilai 0
- *Flip* logic ditulis 0 (sebagai tidak) dan 1 (sebagai ya)
- *Flip* selalu dilakukan sebelum *shift*

Contoh Eksekusi Program (garis bawah menandakan input)

#1

```
Masukkan jumlah elemen: 6
Masukkan elemen linked list pertama: 1 3 5 7 9 11
Masukkan elemen linked list kedua: 2 4 6 8 10 12
Flip linked list kedua: 0
Shift linked list kedua: 0
Hasil: 322
```

#2

```
Masukkan jumlah elemen: 6
Masukkan elemen linked list pertama: 1 3 5 7 9 11
Masukkan elemen linked list kedua: 2 4 6 8 10 12
Flip linked list kedua: 1
Shift linked list kedua: 0
Hasil: 242
(*catatan: diperoleh dari  $1*2+3*12+5*10+7*8+9*6+11*4$ )
```

#3

Masukkan jumlah elemen: 6

Masukkan elemen linked list pertama: 1 3 5 7 9 11

Masukkan elemen linked list kedua: 2 4 6 8 10 12

Flip linked list kedua: 0

Shift linked list kedua: 1

Hasil: 262

(*catatan: diperoleh dari $1*4+3*6+5*8+7*10+9*12+11*2$)