



Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)
Tahun / Semester : 2022-2023 / Genap
Modul : 6 - Linked Lists
Hari, Tanggal Praktikum : Selasa, 21 Maret 2023

Naskah Soal Praktikum

Pembuat Naskah: Syifa Kushirayati, David Azaria Rauf

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 *flowchart* dan *data flow diagram* dari solusi yang anda buat dikumpulkan ke praktikum.stei.itb.ac.id paling lambat pukul 11.00 WIB dua hari kerja setelah sesi praktikum. Tulis pula alasan anda memilih mengerjakan soal yang anda kerjakan dan penjelasan/rancangan dari algoritma yang Anda gunakan!
5. Solusi soal pertama dan kedua harus dapat dikompilasi dengan perintah `make main` dan menghasilkan *file executable* dengan nama `main`.
6. Bila diperlukan, sesuaikanlah isi *Makefile* yang tersedia pada *template repository* untuk memenuhi syarat kompilasi dan *file* keluaran di atas!

Soal 1

Buatlah program untuk mencari *the next lexicographical permutation number* dengan menggunakan struktur data *linked list*. Berikut contoh dari *the next lexicographical permutation number*.

Urutan bilangan	<i>the next lexicographical permutation number</i>
0 1 2 3	0 1 3 2
1 2 3	1 3 2
1 0 0 4	1 0 4 0
4 7 3	7 3 4
0 1 2 5 3 3 0	0 1 3 0 2 3 5
1 0 2 5 0 0	1 0 5 0 0 2

Next lexicographical permutation number dapat dicari dengan mengikuti langkah-langkah berikut.

0. Initial sequence

0	1	2	5	3	3	0
---	---	---	---	---	---	---

1. Find longest non-increasing suffix

0	1	2	5	3	3	0
---	---	---	---	---	---	---

2. Identify pivot

0	1	2	5	3	3	0
---	---	---	---	---	---	---

3. Find rightmost successor to pivot in the suffix

0	1	2	5	3	3	0
---	---	---	---	---	---	---

4. Swap with pivot

0	1	3	5	3	2	0
---	---	---	---	---	---	---

5. Reverse the suffix

0	1	3	0	2	3	5
---	---	---	---	---	---	---

6. Done

0	1	3	0	2	3	5
---	---	---	---	---	---	---

Asumsi yang digunakan:

- Jumlah elemen list yang dimasukkan lebih dari 1.
- Elemen list yang dimasukkan berupa bilangan dari 0/1/2/3/4/5/6/7/8/9.
- Jika elemen list yang dimasukkan berurutan selalu menurun atau konstan, maka program akan menampilkan urutan list yang sama dengan yang dimasukkan. Untuk lebih jelasnya dapat dilihat pada contoh eksekusi program.

***Anda dibebaskan untuk menggunakan template code atau tidak. Asalkan tetap menggunakan struktur data linked list**

Contoh eksekusi program (garis bawah menandakan input):

#1

```
Masukkan jumlah elemen list: 3  
Masukkan elemen list ke-1: 4  
Masukkan elemen list ke-2: 7  
Masukkan elemen list ke-3: 3  
4 7 3 => 7 3 4
```

#2

```
Masukkan jumlah elemen list: 2  
Masukkan elemen list ke-1: 1  
Masukkan elemen list ke-2: 1  
1 1 => 1 1
```

#3

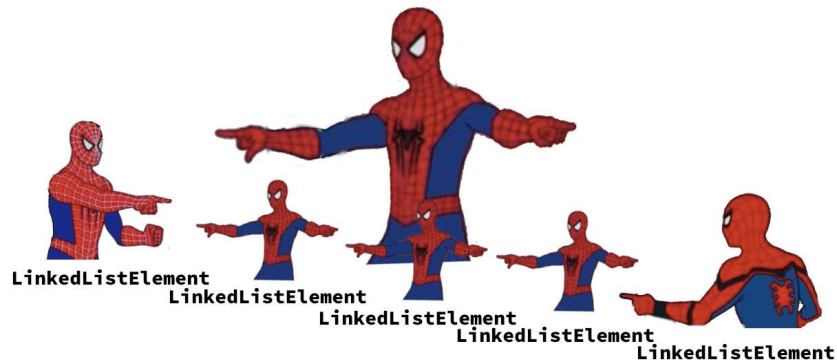
```
Masukkan jumlah elemen list: 5  
Masukkan elemen list ke-1: 5  
Masukkan elemen list ke-2: 4  
Masukkan elemen list ke-3: 3  
Masukkan elemen list ke-4: 2  
Masukkan elemen list ke-5: 1  
5 4 3 2 1 => 5 4 3 2 1
```

#4

```
Masukkan jumlah elemen list: 7  
Masukkan elemen list ke-1: 0  
Masukkan elemen list ke-2: 1  
Masukkan elemen list ke-3: 2  
Masukkan elemen list ke-4: 5  
Masukkan elemen list ke-5: 3  
Masukkan elemen list ke-6: 3  
Masukkan elemen list ke-7: 0  
0 1 2 5 3 3 0 => 0 1 3 0 2 3 5
```

Soal 2

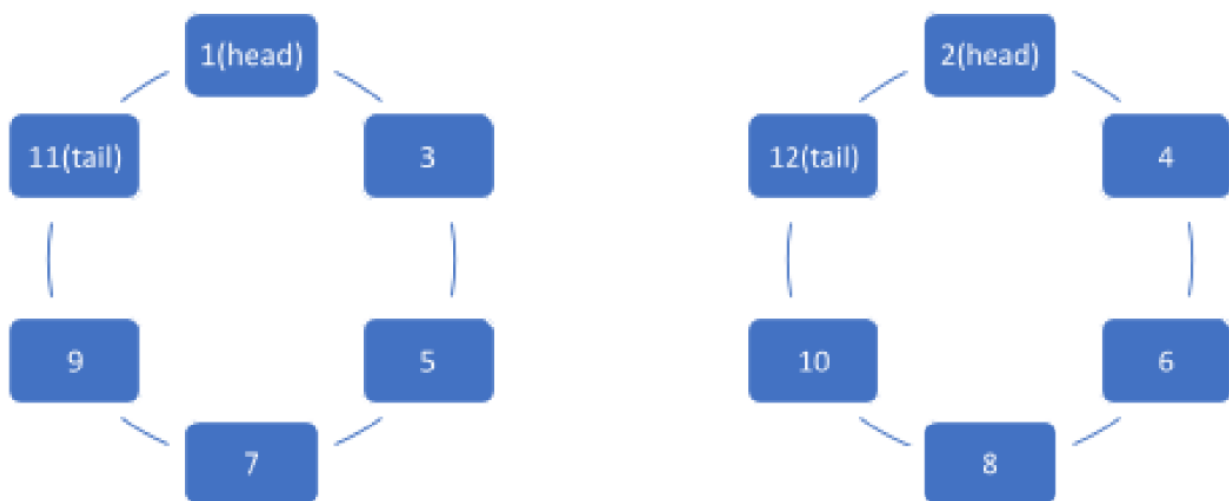
LinkedList



Pada suatu hari, seorang mahasiswa bernama Syauqi diberi tugas untuk melakukan operasi matematika dari dua buah *Circular Linked List* yang berisi bilangan bulat dengan asumsi kedua linked list tersebut memiliki panjang yang selalu sama. Ia ingin membuat program yang dapat menjumlahkan hasil perkalian setiap digit dari kedua linked list tersebut dan menampilkannya pada terminal.

Namun, ia mengalami kesulitan ketika mengimplementasikannya pada program dikarenakan kedua circular linked list tersebut arah putarannya dapat diubah menjadi *clockwise*(0) atau *counterclockwise*(1) sesuai dengan input yang diberikan.

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



$Hasil(clockwise\ clockwise) = 1 * 2 + 3 * 4 + 5 * 6 + 7 * 8 + 9 * 10 + 11 * 12 = 322$

$Hasil(clockwise\ counterclockwise) = 1 * 2 + 3 * 12 + 5 * 10 + 7 * 8 + 9 * 6 + 11 * 4 = 242$

Bantulah Syauqi untuk dapat menyelesaikan masalah tersebut sehingga program yang dapat mengalikan kedua circular linked list dengan benar sesuai dengan arah yang diberikan.

#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
Masukkan arah putaran list pertama: 0
Masukkan arah putaran 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
Masukkan arah putaran list pertama: 0
Masukkan arah putaran list kedua: 1
Hasil: 242

#3

Masukkan jumlah elemen: 2
Masukkan elemen linked list pertama: 1 4
Masukkan elemen linked list kedua: 2 5
Masukkan arah putaran list pertama: 0
Masukkan arah putaran list kedua: 1
Hasil: 22

#4

Masukkan jumlah elemen: 1
Masukkan elemen linked list pertama: 1 4
Masukkan elemen linked list kedua: 2 5
Masukkan arah putaran list pertama: 1
Masukkan arah putaran list kedua: 1
Hasil: 22

#5

Masukkan jumlah elemen: 10
Masukkan elemen linked list pertama: 9 8 7 6 5 4 3 2 1 0
Masukkan elemen linked list kedua: 9 0 1 2 3 4 5 6 7 8
Masukkan arah putaran list pertama: 1
Masukkan arah putaran list kedua: 1
Hasil: 165