
Started on Thursday, 18 November 2021, 5:44 PM

State Finished

Completed on Thursday, 18 November 2021, 7:26 PM

Time taken 1 hour 42 mins

Marks 400.00/400.00

Grade **100.00** out of 100.00

Question **1**

Correct

Mark 300.00 out of 300.00

Time limit

1 s

Memory limit

64 MB

Ambillah ADT List Recursive yang telah Anda kerjakan sebagai tugas pra-praktikum. Tambahkan fungsi dan prosedur berikut ini ke ADT tersebut, simpan dalam file [listrec.h](#) dan **buatlah implementasinya secara rekursif** dan simpan dalam file **listrec.c**. Silahkan menambahkan fungsi/prosedur lain, jika diperlukan.

```

/** Pencarian nilai ekstrim */
/* Prekondisi untuk Max/Min/Sum/Average : List tidak kosong */
ElType maxList (List l);
/* Mengirimkan nilai INFO(p) yang maksimum */

ElType minList (List l);
/* Mengirimkan nilai INFO(p) yang minimum */

ElType sumList (List l);
/* Mengirimkan total jumlah elemen List l */

float averagelist (List l);
/* Mengirimkan nilai rata-rata elemen list l */

/** Operasi-Operasi Lain */
List inverseList (List l);
/* Mengirimkan list baru, hasil invers dari l dengan menyalin semua elemen list.
Semua elemen list baru harus dialokasi */
/* Jika alokasi gagal, hasilnya list kosong */

void splitPosNeg (List l, List *l1, List *l2);
/* I.S. l mungkin kosong */
/* F.S. Berdasarkan l, dibentuk dua buah list l1 dan l2 */
/* l1 berisi semua elemen l yang positif atau 0, sedangkan l2 berisi
semua elemen l yang negatif; semua dengan urutan yang sama seperti di l */
/* l tidak berubah: Semua elemen l1 dan l2 harus dialokasi */
/* Jika l kosong, maka l1 dan l2 kosong */

void splitOnX (List l, ElType x, List *l1, List *l2);
/* I.S. l dan x terdefinisi, l1 dan l2 sembarang. */
/* F.S. l1 berisi semua elemen l yang lebih kecil dari x, dengan urutan
kemunculan yang sama, l2 berisi semua elemen l yang tidak masuk ke
l1, dengan urutan kemunculan yang sama. */

int compareList (List l1, List l2);
/* Menghasilkan: -1 jika l1 < l2, 0 jika l1 = l2, dan 1 jika l1 > l2 */
/* Jika l[i] adalah elemen l pada urutan ke-i dan |l| adalah panjang l: */
/* l1 = l2: |l1| = |l2| dan untuk semua i, l1[i] = l2[i] */
/* l1 < l2: Jika i adalah urutan elemen yang berbeda yang terkecil
dari l1 dan l2, l1[i]<l2[i] atau: Jika pada semua elemen pada
urutan i yang sama, l1[i]=l2[i], namun |l1|<|l2| */
/* Contoh: [3,5,6,7] < [4,4,5,6]; [1,2,3]<[1,2,3,4] */
/* l1>l2: kebalikan dari l1<l2 */

boolean isAllExist (List l1, List l2);
/* Menghasilkan true jika semua elemen dalam l1 terdapat dalam l2 (tanpa
memperhatikan urutan ataupun banyaknya elemen).
Kedua list mungkin kosong. Jika l1 kosong, maka hasilnya false. */

```

Submit file **listrec.c**

C

 [listrec.c](#)

Score: 300

Blackbox

Score: 300

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.64 MB
2	10	Accepted	0.00 sec, 1.60 MB
3	10	Accepted	0.00 sec, 1.71 MB
4	10	Accepted	0.00 sec, 1.49 MB
5	10	Accepted	0.00 sec, 1.71 MB
6	10	Accepted	0.00 sec, 1.63 MB
7	10	Accepted	0.00 sec, 1.71 MB
8	10	Accepted	0.00 sec, 1.71 MB
9	10	Accepted	0.00 sec, 1.63 MB
10	10	Accepted	0.00 sec, 1.68 MB
11	10	Accepted	0.00 sec, 1.71 MB
12	10	Accepted	0.00 sec, 1.64 MB
13	10	Accepted	0.00 sec, 1.64 MB
14	10	Accepted	0.00 sec, 1.64 MB
15	10	Accepted	0.00 sec, 1.64 MB
16	10	Accepted	0.00 sec, 1.63 MB
17	10	Accepted	0.00 sec, 1.60 MB
18	10	Accepted	0.00 sec, 1.54 MB
19	10	Accepted	0.00 sec, 1.62 MB
20	10	Accepted	0.00 sec, 1.66 MB
21	10	Accepted	0.00 sec, 1.54 MB
22	10	Accepted	0.00 sec, 1.66 MB
23	10	Accepted	0.00 sec, 1.64 MB
24	10	Accepted	0.00 sec, 1.66 MB
25	10	Accepted	0.00 sec, 1.71 MB
26	10	Accepted	0.00 sec, 1.57 MB
27	10	Accepted	0.00 sec, 1.64 MB
28	10	Accepted	0.00 sec, 1.64 MB
29	10	Accepted	0.00 sec, 1.64 MB
30	10	Accepted	0.00 sec, 1.63 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit

1 s

Memory limit

64 MB

Mr. Jek mengadakan sebuah acara undian berhadiah yang didatangi oleh banyak peserta. Karena jumlah peserta yang datang terlalu banyak, Mr. Jek akan melakukan eliminasi terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Setiap peserta undian dengan nomor undian k akan dimasukkan ke dalam list kandidat antara paling depan atau paling belakang list. List kandidat dapat dikurangi dengan 3 cara, yaitu mengambil elemen pada indeks ganjil, indeks genap, atau mengambil peserta di depan peserta dengan nomor undian X . List juga dapat diputar urutannya. Berikut adalah operasi yang dapat dilakukan pada list kandidat:

- **F k** : Memasukkan k ke elemen pertama list
- **B k** : Memasukkan k ke elemen terakhir list
- **O** : Mengambil elemen-elemen dengan indeks ganjil dan membentuk list baru (HEAD adalah indeks 1)
- **E** : Mengambil elemen-elemen dengan indeks genap dan membentuk list baru (HEAD adalah indeks 1)
- **R** : Memutar urutan elemen pada list
- **P x** : Mengambil elemen-elemen di depan elemen dengan nomor undian x pada list. Jika x tidak ada dalam list, maka list tidak berubah
- **X** : Operasi selesai

Mr. Jek sangat teliti sehingga dipastikan tidak ada nomor undian yang sama. Setelah operasi selesai, tentukanlah nomor undian yang beruntung yang akan lolos ke babak berikutnya, sesuai dengan urutan pada list. Jika tidak ada, tuliskan "Semua gugur".

Contoh

Input	Output	Keterangan
X	Semua gugur	
B 1 B 2 B 3 B 4 B 5 B 6 B 7 O R E X	5 1	Hingga B 7, list menjadi [1,2,3,4,5,6,7] Pada O, list menjadi [1,3,5,7] Pada R, list menjadi [7,5,3,1] Pada E, list menjadi [5,1]
F 1 F 2 B 3 F 4 B 5 B 6 F 7 P 1 R F 8 O X	8 4	Hingga F 7, list menjadi [7,4,2,1,3,5,6] Pada P 1, list menjadi [7,4,2] Pada R, list menjadi [2,4,7] Pada F 8, list menjadi [8,2,4,7] Pada O, list menjadi [8,4]

Kerjakan dan kumpulkan dalam file **undian.c**

Petunjuk:

Untuk membantu, definisikan dan implementasikan prosedur-prosedur berikut pada **undian.c** Anda:

```

List splitOddIndex(List l);
/* Mengembalikan sebuah list baru yang elemennya adalah elemen-elemen pada indeks
ganjil pada list l, sesuai dengan urutannya. Elemen HEAD merupakan indeks 1.
Jika tidak ada elemen yang bisa diambil, kembalikan NULL */

List splitEvenIndex(List l);
/* Mengembalikan sebuah list baru yang elemennya adalah elemen-elemen pada indeks
genap pada list l, sesuai dengan urutannya. Elemen HEAD merupakan indeks 1.
Jika tidak ada elemen yang bisa diambil, kembalikan NULL */

List splitBeforeX(List l, ElType x);
/* Mengembalikan sebuah list baru yang elemennya adalah elemen-elemen di depan elemen
dengan nilai x pada list l, sesuai dengan urutannya. Jika x tidak ada pada list, maka
kembalikan list l.
Jika tidak ada elemen yang bisa diambil, kembalikan NULL */

List reverseList(List l);
/* Mengembalikan sebuah list baru yang urutannya merupakan kebalikan dari list l.
Jika l kosong, kembalikan list kosong. */

```

C

 [undian.c](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.52 MB
2	10	Accepted	0.00 sec, 1.52 MB
3	10	Accepted	0.00 sec, 1.58 MB
4	10	Accepted	0.00 sec, 1.50 MB
5	10	Accepted	0.00 sec, 1.54 MB
6	10	Accepted	0.00 sec, 1.52 MB
7	10	Accepted	0.00 sec, 1.47 MB
8	10	Accepted	0.00 sec, 1.49 MB
9	10	Accepted	0.00 sec, 1.57 MB
10	10	Accepted	0.00 sec, 1.54 MB

◀ Pra Praktikum 12

Jump to...

bintree.h ▶