

[Dashboard](#) / [My courses](#) / [ITB IF2110 1 2223](#) / [Praktikum 11](#) / [Latihan Praktikum 11](#)

Started on	Thursday, 17 November 2022, 10:26 PM
State	Finished
Completed on	Thursday, 17 November 2022, 10:30 PM
Time taken	3 mins 57 secs
Marks	650.00/650.00
Grade	100.00 out of 100.00

Question **1**

Correct

Mark 230.00 out of 230.00

Time limit	1 s
Memory limit	64 MB

Kumpulkan **listrec.c**

C

 [listrec.c](#)

Score: 230

Blackbox

Score: 230

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.60 MB
2	10	Accepted	0.00 sec, 1.64 MB
3	10	Accepted	0.00 sec, 1.60 MB
4	10	Accepted	0.00 sec, 1.62 MB
5	10	Accepted	0.00 sec, 1.71 MB
6	10	Accepted	0.00 sec, 1.51 MB
7	10	Accepted	0.00 sec, 1.62 MB
8	10	Accepted	0.00 sec, 1.71 MB
9	10	Accepted	0.00 sec, 1.51 MB
10	10	Accepted	0.00 sec, 1.58 MB
11	10	Accepted	0.00 sec, 1.62 MB
12	10	Accepted	0.00 sec, 1.63 MB
13	10	Accepted	0.00 sec, 1.50 MB
14	10	Accepted	0.00 sec, 1.66 MB
15	10	Accepted	0.00 sec, 1.60 MB
16	10	Accepted	0.00 sec, 1.51 MB
17	10	Accepted	0.00 sec, 1.71 MB
18	10	Accepted	0.00 sec, 1.50 MB
19	10	Accepted	0.00 sec, 1.63 MB
20	10	Accepted	0.00 sec, 1.65 MB
21	10	Accepted	0.00 sec, 1.67 MB
22	10	Accepted	0.00 sec, 1.71 MB
23	10	Accepted	0.00 sec, 1.64 MB

Question **2**

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**. Silakan 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, 1.49 MB
3	10	Accepted	0.00 sec, 1.49 MB
4	10	Accepted	0.00 sec, 1.51 MB
5	10	Accepted	0.00 sec, 1.64 MB
6	10	Accepted	0.00 sec, 1.60 MB
7	10	Accepted	0.00 sec, 1.60 MB
8	10	Accepted	0.00 sec, 1.52 MB
9	10	Accepted	0.00 sec, 1.66 MB
10	10	Accepted	0.00 sec, 1.66 MB
11	10	Accepted	0.00 sec, 1.68 MB
12	10	Accepted	0.00 sec, 1.66 MB
13	10	Accepted	0.00 sec, 1.65 MB
14	10	Accepted	0.00 sec, 1.71 MB
15	10	Accepted	0.00 sec, 1.63 MB
16	10	Accepted	0.00 sec, 1.55 MB
17	10	Accepted	0.00 sec, 1.54 MB
18	10	Accepted	0.00 sec, 1.55 MB
19	10	Accepted	0.00 sec, 1.65 MB
20	10	Accepted	0.00 sec, 1.66 MB
21	10	Accepted	0.00 sec, 1.65 MB
22	10	Accepted	0.00 sec, 1.64 MB
23	10	Accepted	0.00 sec, 1.64 MB
24	10	Accepted	0.00 sec, 1.54 MB
25	10	Accepted	0.00 sec, 1.64 MB
26	10	Accepted	0.00 sec, 1.71 MB
27	10	Accepted	0.00 sec, 1.64 MB
28	10	Accepted	0.00 sec, 1.60 MB
29	10	Accepted	0.00 sec, 1.60 MB
30	10	Accepted	0.00 sec, 1.55 MB

Question **3**

Correct

Mark 120.00 out of 120.00

Time limit	15 s
Memory limit	64 MB

Tuan Cel merupakan guru dari sebuah kelas komputer pada sekolah BNMO. Pada suatu hari, Tuan Cel ingin merilis sebuah tugas proyek yang akan dikerjakan secara kelompok. Cara membagikan kelompok Tuan Cel cukup unik, tuan cel ingin me-random murid-murid pada suatu kelompok sehingga mereka dapat mengenal satu sama lain.

Awalnya Tuan Cel menyuruh murid-murid untuk membentuk suatu barisan, lalu akan dibentuk kelompok berdasarkan baris tersebut. Saat berbaris, murid-murid yang sudah saling mengenal pasti berusaha untuk berbaris secara berurutan agar kesempatan untuk berada di satu kelompok dapat lebih besar, namun Tuan Cel lebih pintar, Tuan Cel membagi barisan tersebut menjadi beberapa partisi, dan setiap partisi tersebut berukuran K. Lalu setiap partisi tersebut akan di-reverse sehingga barisan tersebut akan menjadi teracak.

Tetapi setelah di-reverse Tuan Cel ingin mengetahui hasil dari barisan yang telah diacak tersebut, karena murid Tuan Cel cukup banyak, kalian diminta untuk membuat program yang mampu menghasilkan keadaan barisan diatas, bantulah Tuan Cel!

Input program pada barisan pertama berupa berapa panjang dari barisan murid Tuan Cel dan K (panjang partisi), lalu untuk baris selanjutnya berisi input elemen-elemen dari list. Untuk output dari program sendiri berupa barisan yang telah dijelaskan sebelumnya.

Gunakan ADT List Recursive yang telah Anda kerjakan pada pra praktikum. Lalu kumpulkan **reverse.c**

Input	Output	Keterangan
6 3 1 2 3 4 5 6	3 2 1 6 5 4	-
7 4 4 5 6 7 8 9 3	7 6 5 4 3 9 8	-

Catatan:

Harus dikerjakan secara rekursif, kode program akan dicek manual oleh asisten, dan jika terbukti menggunakan prosedural nilai untuk soal ini akan menjadi 0

C

 [reverse.c](#)

Score: 150

Blackbox

Score: 150

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.67 MB
2	10	Accepted	0.00 sec, 1.65 MB
3	10	Accepted	0.00 sec, 1.64 MB
4	10	Accepted	0.00 sec, 1.50 MB
5	10	Accepted	0.00 sec, 1.67 MB
6	10	Accepted	0.00 sec, 1.55 MB

6	10	Accepted	0.00 sec, 1.55 MB
No	Score	Verdict	Description
7	10	Accepted	0.00 sec, 1.67 MB
8	10	Accepted	0.00 sec, 1.65 MB
9	10	Accepted	0.00 sec, 1.63 MB
10	10	Accepted	0.00 sec, 1.67 MB
11	10	Accepted	0.00 sec, 1.75 MB
12	10	Accepted	0.00 sec, 1.70 MB
13	10	Accepted	4.75 sec, 6.22 MB
14	10	Accepted	4.60 sec, 6.29 MB
15	10	Accepted	4.67 sec, 6.23 MB

[◀ listrec.h](#)

Jump to...