
Started on Sunday, 18 September 2022, 10:34 PM

State Finished

Completed on Monday, 19 September 2022, 12:18 AM

Time taken 1 hour 44 mins

Marks 1370.00/1370.00

Grade **100.00** out of 100.00

Question **1**

Correct

Mark 490.00 out of 490.00

Time limit

1 s

Memory limit

64 MB

Downloadlah file header [liststatik.h](#) yang merupakan ADT List dengan representasi implisit, alokasi statik dengan elemen bilangan bulat positif, lalu buatlah file implementasinya. Untuk menandai elemen yang tidak terdefinisi dalam list, digunakan nilai -9999. Kumpulkan file bernama **liststatik.c**

C

 [liststatik.c](#)

Score: 490

Blackbox

Score: 490

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.58 MB
2	10	Accepted	0.00 sec, 1.59 MB
3	10	Accepted	0.00 sec, 1.64 MB
4	10	Accepted	0.00 sec, 1.49 MB
5	10	Accepted	0.00 sec, 1.49 MB
6	10	Accepted	0.00 sec, 1.66 MB
7	10	Accepted	0.00 sec, 1.55 MB
8	10	Accepted	0.00 sec, 1.61 MB
9	10	Accepted	0.00 sec, 1.58 MB
10	10	Accepted	0.00 sec, 1.57 MB
11	10	Accepted	0.00 sec, 1.54 MB
12	10	Accepted	0.00 sec, 1.62 MB
13	10	Accepted	0.00 sec, 1.56 MB
14	10	Accepted	0.00 sec, 1.61 MB
15	10	Accepted	0.00 sec, 1.50 MB
16	10	Accepted	0.00 sec, 1.57 MB
17	10	Accepted	0.00 sec, 1.55 MB
18	10	Accepted	0.00 sec, 1.65 MB
19	10	Accepted	0.00 sec, 1.58 MB
20	10	Accepted	0.00 sec, 1.71 MB
21	10	Accepted	0.00 sec, 1.57 MB
22	10	Accepted	0.00 sec, 1.67 MB

No	Score	Verdict	Description
23	10	Accepted	0.00 sec, 1.63 MB

24	10	Accepted	0.00 sec, 1.66 MB
25	10	Accepted	0.00 sec, 1.54 MB
26	10	Accepted	0.00 sec, 1.65 MB
27	10	Accepted	0.00 sec, 1.66 MB
28	10	Accepted	0.00 sec, 1.71 MB
29	10	Accepted	0.00 sec, 1.61 MB
30	10	Accepted	0.00 sec, 1.61 MB
31	10	Accepted	0.00 sec, 1.50 MB
32	10	Accepted	0.00 sec, 1.63 MB
33	10	Accepted	0.00 sec, 1.61 MB
34	10	Accepted	0.00 sec, 1.65 MB
35	10	Accepted	0.00 sec, 1.64 MB
36	10	Accepted	0.00 sec, 1.67 MB
37	10	Accepted	0.00 sec, 1.67 MB
38	10	Accepted	0.00 sec, 1.64 MB
39	10	Accepted	0.00 sec, 1.59 MB
40	10	Accepted	0.00 sec, 1.64 MB
41	10	Accepted	0.00 sec, 1.65 MB
42	10	Accepted	0.00 sec, 1.60 MB
43	10	Accepted	0.00 sec, 1.50 MB
44	10	Accepted	0.00 sec, 1.72 MB
45	10	Accepted	0.00 sec, 1.71 MB
46	10	Accepted	0.00 sec, 1.67 MB
47	10	Accepted	0.00 sec, 1.56 MB
48	10	Accepted	0.00 sec, 1.66 MB
49	10	Accepted	0.00 sec, 1.57 MB

Question **2**
Correct
Mark 580.00 out of 580.00

Time limit	1 s
Memory limit	64 MB

Downloadlah file header [listdin.h](#) yang merupakan ADT list dengan representasi eksplisit dan alokasi dinamik. Implementasikan dalam ADT tersebut dalam file bernama listdin.c. Kumpulkan file Bernama **listdin.c**

C

 [listdin.c](#)

Score: 580

Blackbox

Score: 580

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.55 MB
2	10	Accepted	0.00 sec, 1.65 MB
3	10	Accepted	0.00 sec, 1.50 MB
4	10	Accepted	0.00 sec, 1.60 MB
5	10	Accepted	0.00 sec, 1.61 MB
6	10	Accepted	0.00 sec, 1.65 MB
7	10	Accepted	0.00 sec, 1.61 MB
8	10	Accepted	0.00 sec, 1.65 MB
9	10	Accepted	0.00 sec, 1.61 MB
10	10	Accepted	0.00 sec, 1.47 MB
11	10	Accepted	0.00 sec, 1.55 MB
12	10	Accepted	0.00 sec, 1.61 MB
13	10	Accepted	0.00 sec, 1.58 MB
14	10	Accepted	0.00 sec, 1.57 MB
15	10	Accepted	0.00 sec, 1.61 MB
16	10	Accepted	0.00 sec, 1.60 MB
17	10	Accepted	0.00 sec, 1.62 MB
18	10	Accepted	0.00 sec, 1.58 MB
19	10	Accepted	0.01 sec, 1.57 MB
20	10	Accepted	0.00 sec, 1.57 MB
21	10	Accepted	0.00 sec, 1.66 MB
22	10	Accepted	0.00 sec, 1.55 MB
23	10	Accepted	0.02 sec, 1.54 MB

No	Score	Verdict	Description
----	-------	---------	-------------

24	10	Accepted	0.00 sec, 1.60 MB
25	10	Accepted	0.00 sec, 1.60 MB
26	10	Accepted	0.00 sec, 1.54 MB
27	10	Accepted	0.00 sec, 1.60 MB
28	10	Accepted	0.00 sec, 1.61 MB
29	10	Accepted	0.00 sec, 1.71 MB
30	10	Accepted	0.00 sec, 1.65 MB
31	10	Accepted	0.00 sec, 1.51 MB
32	10	Accepted	0.00 sec, 1.67 MB
33	10	Accepted	0.00 sec, 1.67 MB
34	10	Accepted	0.00 sec, 1.56 MB
35	10	Accepted	0.00 sec, 1.64 MB
36	10	Accepted	0.00 sec, 1.57 MB
37	10	Accepted	0.00 sec, 1.65 MB
38	10	Accepted	0.00 sec, 1.67 MB
39	10	Accepted	0.00 sec, 1.67 MB
40	10	Accepted	0.00 sec, 1.65 MB
41	10	Accepted	0.00 sec, 1.71 MB
42	10	Accepted	0.00 sec, 1.55 MB
43	10	Accepted	0.01 sec, 1.66 MB
44	10	Accepted	0.00 sec, 1.50 MB
45	10	Accepted	0.02 sec, 1.59 MB
46	10	Accepted	0.00 sec, 1.57 MB
47	10	Accepted	0.00 sec, 1.50 MB
48	10	Accepted	0.00 sec, 1.68 MB
49	10	Accepted	0.00 sec, 1.65 MB
50	10	Accepted	0.00 sec, 1.67 MB
51	10	Accepted	0.00 sec, 1.72 MB
52	10	Accepted	0.00 sec, 1.50 MB
53	10	Accepted	0.01 sec, 1.67 MB
54	10	Accepted	0.00 sec, 1.71 MB
55	10	Accepted	0.00 sec, 1.64 MB
56	10	Accepted	0.00 sec, 1.71 MB
57	10	Accepted	0.00 sec, 1.65 MB
58	10	Accepted	0.00 sec, 1.65 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit

1 s

Memory limit

64 MB

Dengan memanfaatkan ADT List Statik yang telah Anda buat pada tugas pra-praktikum, buatlah sebuah program yang digunakan untuk membaca sebuah List, misalnya T, dengan ukuran efektif N (N dibaca dari keyboard), lalu membaca elemen-elemen T dari keyboard. N harus bernilai $0 \leq N \leq \text{MaxNbEl}(T)$. Kemudian program menuliskan ulang isi T dalam format: $[e_1, e_2, \dots, e_n]$. Selanjutnya, program menerima masukan sebuah nilai integer, misalnya X dan menampilkan beberapa hal sebagai berikut secara berturut-turut:

- Berapa banyak kemunculan X dalam array.
- Jika X ada di array, tuliskan di indeks ke-berapa X muncul terakhir kali. Jika X tidak ada di array, tuliskan "X tidak ada".
- Apakah X adalah nilai maksimum, nilai minimum, atau nilai tengah (median) dari deret nilai dalam array (lihat contoh interaksi di bawah), jika X ada di array.

Gunakan **fungsi dan prosedur** yang telah kalian buat pada pra-praktikum, include file .h yang bersangkutan.

Median adalah nilai yang dalam rangking (terkecil ke terbesar) dari seluruh nilai mahasiswa yang valid merupakan nilai yang tepat berada di tengah. Jika banyaknya mahasiswa N dan N adalah ganjil, maka median adalah nilai pada rangking ke- $(N \div 2) + 1$. Jika N genap, maka nilai tengah adalah nilai pada rangking ke- $(N \div 2)$. Petunjuk: Gunakan prosedur untuk sorting.

Dipersilakan membuat fungsi/prosedur tambahan dalam program ini jika perlu.

Kumpulkan file bernama olist.c dan liststatik.c (file dari pra-praktikum kalian) dalam zip bernama **olist.zip**

Hint (tidak wajib diikuti, untuk kemudahan saja): Buat fungsi/prosedur untuk invers list atau mengubah urutan, misalnya $[1, 6, 3]$ menjadi $[3, 6, 1]$. Gunakan fungsi indexOf dan inverse untuk mendapatkan X indeks terakhir.

Contoh input/output:

Input	Output	Keterangan
6 1 -1 7 4 -1 3 -1	[1,-1,7,4,-1,3] 2 4 minimum	N = 6 Isi array = [1,-1,7,4,-1,3] X = -1 Muncul terakhir kali di indeks = 4 maksimum = 7 minimum = -1 median = 1
6 1 -1 2 4 -1 2 8	[1,-1,2,4,-1,2] 0 8 tidak ada	N = 6 Isi array = [1,-1,2,4,-1,2] X = 8 X tidak ada dalam array

5		N = 5
3	[3,2,2,4,2]	Isi array = {3,2,2,4,2}
2	3	X = 2
2	4	Muncul terakhir kali di indeks = 4
4	minimum	maksimum = 4
2	median	minimum = 2
2		median = 2
5		N = 5
1	[1,1,1,1,1]	Isi array = [1,1,1,1,1]
1	5	X = 1
1	4	Muncul terakhir kali di indeks = 4
1	maksimum	maksimum = 1
1	minimum	minimum = 1
1	median	median = 1
1		

C

 [olist.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.63 MB
2	10	Accepted	0.00 sec, 1.54 MB
3	10	Accepted	0.00 sec, 1.50 MB
4	10	Accepted	0.00 sec, 1.56 MB
5	10	Accepted	0.00 sec, 1.65 MB
6	10	Accepted	0.00 sec, 1.57 MB
7	10	Accepted	0.00 sec, 1.52 MB
8	10	Accepted	0.00 sec, 1.63 MB
9	10	Accepted	0.00 sec, 1.57 MB
10	10	Accepted	0.00 sec, 1.57 MB

Question **4**

Correct

Mark 100.00 out of 100.00

Time limit

1 s

Memory limit

64 MB

Tuan Rez adalah seorang koordinator praktikum. Saat menyusun jadwal, ia menemukan beberapa NIM praktikan muncul di beberapa kelas yang berbeda. Seharusnya, seorang praktikan hanya muncul sekali pada salah satu kelas. Ayo bantu Tuan Rez untuk menemukan NIM yang muncul lebih dari satu kali pada kelas yang berbeda. Gunakan List Statik yang telah dibuat pada pra-praktikum. Problem ini dapat diselesaikan dengan list intersection, silakan implementasikan list intersection untuk menyelesaikan persoalan ini.

Beberapa informasi yang perlu diketahui:

- NIM pada kelas tidak selalu terurut.
- Input awal adalah jumlah orang pada kelas pertama (N), lalu diikuti dengan N baris NIM dari kelas pertama. Lalu, diikuti jumlah orang pada kelas kedua (M), lalu diikuti dengan M baris NIM dari kelas kedua.
- NIM selalu bertipe bilangan bulat (integer).
- NIM bisa terdiri dari 1 sampai 8 digit.
- NIM tidak mungkin berawalan 0. (Contoh yang tidak mungkin: 0001)
- NIM hanya bisa muncul satu kali pada kelas yang sama.
- Kelas bisa kosong, jika kosong, berarti tidak ada yang akan sama.
- Output harap disort dari NIM terkecil ke NIM terbesar.
- Kelas hanya ada dua (2).

Hint (tidak wajib, untuk kemudahan saja): Gunakan indexOf, printList, isEmpty, listLength, readList, insertList, sortList yang telah kalian buat pada pra-praktikum.

Kumpulkan file Bernama intersect.c dan liststatik.c (file dari pra-praktikum kalian) dalam zip bernama **intersect.zip**

Contoh input/output:

Input	Output	Keterangan
3 13522001 13522002 13522003		N = 3 Isi array kelas 1 = [13522001, 13522002, 13522003]
4 13522002 13522004 13522005 13522006	1 [13522002]	M = 3 Isi array kelas 2 = [13522002, 13522004, 13522005, 13522006] Hanya 13522002 yang muncul dua kali di dua kelas berbeda.
0 2 13519001 13519002	0 []	N = 0 Isi array kelas 1 = [] M = 2 Isi array kelas 2 = [13519001, 13519002] Tidak ada yang sama.
0 0	0 []	Setiap kelas kosong.

1 13521001 1 13521002	0 []	N = 1 Isi array kelas 1 = [13521001] M = 1 Isi array kelas 2 = [13521002] Tidak ada yang sama.
6 13519001 13518077 18218182 13518013 13519003 13522001 5 13518013 18218182 18218001 13519001 13522002	3 [13518013,13519001,18218182]	N = 6 Isi array kelas 1 = [13519001, 13518077, 18218182, 13518013, 13519003, 13522001] M = 5 Isi array kelas 2 = [13518013, 18218182, 18218001, 13519001, 13522002] Ada 3 NIM yang muncul dua kali di dua kelas berbeda.

C

 [intersect.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.54 MB
2	10	Accepted	0.00 sec, 1.61 MB
3	10	Accepted	0.00 sec, 1.61 MB
4	10	Accepted	0.00 sec, 1.50 MB
5	10	Accepted	0.00 sec, 1.64 MB
6	10	Accepted	0.00 sec, 1.68 MB
7	10	Accepted	0.00 sec, 1.68 MB
8	10	Accepted	0.00 sec, 1.63 MB
9	10	Accepted	0.00 sec, 1.64 MB
10	10	Accepted	0.00 sec, 1.71 MB

Question **5**

Correct

Mark 100.00 out of 100.00

Time limit

1 s

Memory limit

64 MB

Tuan Cel tiba-tiba sakit dan tidak masuk, sehingga Tuan Rez harus menggabungkan dua kelas praktikum menjadi satu. Tuan Rez membutuhkan list NIM Mahasiswa yang terurut dari penggabungan dua kelas tersebut. Namun, data NIM yang dimiliki Tuan Rez tidak rapih dan terdapat NIM yang sama pada dua kelas praktikum tersebut. Sedangkan yang diinginkan adalah NIM gabungan dari dua kelas tersebut tanpa ada yang duplikat. Gunakan ADT List Dinamik yang telah kalian implementasikan pada Pra-Praktikum. Permasalahan ini bisa diselesaikan dengan menggabungkan kedua array menjadi satu tanpa elemen duplikat, lalu di sort.

Beberapa informasi yang perlu diketahui:

- NIM pada kelas tidak selalu terurut.
- Input awal adalah jumlah orang pada kelas pertama (N), lalu diikuti dengan N baris NIM dari kelas pertama. Lalu, diikuti jumlah orang pada kelas kedua (M), lalu diikuti dengan M baris NIM dari kelas kedua.
- NIM selalu bertipe bilangan bulat (integer).
- NIM bisa terdiri dari 1 sampai 8 digit.
- NIM tidak mungkin berawalan 0. (Contoh yang tidak mungkin: 0001)
- NIM hanya bisa muncul satu kali pada kelas yang sama.
- Kelas bisa kosong.
- Output harap disort dari NIM terkecil ke NIM terbesar.
- Kelas hanya ada dua (2).
- Kelas bisa memiliki data NIM yang sama.

Kumpulkan file Bernama merge.c dan listdin.c (file dari pra-praktikum kalian) dalam zip bernama **merge.zip**

Hint: gunakan fungsi/prosedur (create list, read list, dst.) yang telah kalian implementasikan pada pra-praktikum.

Sebelum memanggil fungsi readList, silakan gunakan CreateListDin dengan kapasitas 200. List hasil menggunakan kapasitas 400.

Contoh input/output:

Input	Output	Keterangan
3 1 3 2 4 3 2 6 5	5 [1,2,3,5,6]	N = 3 Kelas 1 = [1,3,2] M = 4 Kelas 2 = [3,2,6,5]
1 109 2 117 189	3 [109,117,189]	N = 1 Kelas 1 = [109] M = 2 Kelas 2 = [117, 189]
0 2 13519002 13519049	2 [13519002,13519049]	N = 0 Kelas 1 = [] M = 2

		Kelas 2 = [13519002,13519049]
0 0	0 []	N = 0, M = 0
2 110 128 2 110 128	2 [110,128]	N = 2 Kelas 1 = [110, 128] M = 2 Kelas 2 = [110, 128]

C

 [merge.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.49 MB
2	10	Accepted	0.00 sec, 1.64 MB
3	10	Accepted	0.00 sec, 1.56 MB
4	10	Accepted	0.00 sec, 1.61 MB
5	10	Accepted	0.00 sec, 1.57 MB
6	10	Accepted	0.00 sec, 1.63 MB
7	10	Accepted	0.00 sec, 1.52 MB
8	10	Accepted	0.00 sec, 1.64 MB
9	10	Accepted	0.00 sec, 1.65 MB
10	10	Accepted	0.00 sec, 1.68 MB

[◀ listdin.h](#)

Jump to...

[matrix.h ▶](#)