

Started on	Thursday, 10 October 2024, 9:28 PM
State	Finished
Completed on	Friday, 11 October 2024, 12:42 AM
Time taken	3 hours 13 mins
Marks	300.00/300.00
Grade	10.00 out of 10.00 (100%)

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Barisan Pak Asep Spakbor

Deskripsi

Pak Asep Spakbor ingin menyusun barisan N siswa berdasarkan tinggi badan, dari yang terpendek di depan hingga tertinggi di belakang. Setiap siswa baru akan mencari tempat di barisan dengan berdiri di depan siswa yang lebih tinggi. Jika tidak ada yang lebih tinggi, siswa akan berdiri di belakang. Setiap kali siswa baru bergabung, siswa di belakangnya harus mundur untuk memberi ruang.

Tugas Anda adalah menulis program yang menghitung total langkah mundur yang terjadi selama proses ini.

Format Masukan

- Baris pertama berisi sebuah bilangan bulat N ($1 \leq N \leq 1000$).
- Baris kedua berisi N bilangan bulat unik (1 - 1000) yang merepresentasikan tinggi siswa.

Format Keluaran

Jumlah total langkah mundur yang terjadi.

Contoh

No	Masukan	Keluaran
1.	3 4 5 1	2

Penjelasan

Terdapat 3 siswa dengan tinggi 4, 5, dan 1. Berikut adalah cara mereka masuk ke dalam barisan:

1. Siswa dengan tinggi 4 masuk pertama kali. Karena barisan kosong, dia langsung berdiri tanpa ada langkah mundur. Barisan: [4]
2. Siswa dengan tinggi 5 masuk berikutnya. Karena tidak ada yang lebih tinggi darinya, dia berdiri di belakang siswa dengan tinggi 4. Tidak ada yang mundur. Barisan: [4, 5]
3. Siswa dengan tinggi 1 masuk terakhir. Karena dia lebih pendek dari semua siswa di barisan, dia harus berdiri paling depan. Untuk itu, siswa dengan tinggi 4 dan 5 harus mundur masing-masing satu langkah agar ada ruang di depannya. Barisan: [1, 4, 5]

Jadi, total ada 2 langkah mundur (satu dari siswa tinggi 4 dan satu lagi dari siswa tinggi 5) dalam proses ini.

Catatan

- Keluaran diakhiri dengan newline (\n)
- Download dan gunakan file [boolean.h](#), [listdin.h](#) & [listdin.c](#) yang telah dibuat sebelumnya
- Download dan kumpulkan jawaban dalam file bernama [barisan.c](#)

C

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, 1.65 MB
3	10	Accepted	0.00 sec, 1.55 MB

No	Score	Verdict	Description
4	10	Accepted	0.00 sec, 1.64 MB
5	10	Accepted	0.00 sec, 1.57 MB
6	10	Accepted	0.00 sec, 1.64 MB
7	10	Accepted	0.00 sec, 1.63 MB
8	10	Accepted	0.00 sec, 1.60 MB
9	10	Accepted	0.00 sec, 1.60 MB
10	10	Accepted	0.00 sec, 1.61 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Pivot

Deskripsi

Sebuah algoritma **Partisi** $O(n)$ digunakan untuk membagi sebuah list L dengan memilih sebuah elemen sebagai **pivot** (yang merupakan salah satu elemen dari L) menjadi tiga bagian:

- 1. Sub-list kiri yang berisi elemen-elemen yang lebih kecil atau sama dengan pivot.
- 2. Pivot itu sendiri.
- 3. Sub-list kanan yang berisi elemen-elemen yang lebih besar dari pivot.

Algoritma Partisi merupakan bagian penting dari algoritma pengurutan **Quicksort**. Biasanya pemilihan pivot dilakukan secara acak, sehingga Quicksort memiliki kompleksitas waktu $O(n \log n)$.

Permasalahan dalam soal ini adalah sebagai berikut: Diberikan sebuah list L yang berisi N bilangan bulat yang semua elemennya berbeda satu sama lain. Kita melakukan proses Partisi pada L dengan menggunakan salah satu elemen dari L sebagai pivot untuk menghasilkan list yang telah diubah yaitu L' . Berdasarkan L' ini, tugas Anda adalah menghitung berapa banyak elemen yang mungkin bisa dipilih sebagai pivot untuk menghasilkan bentuk list seperti L' tersebut.

Format Masukan

- Baris pertama berisi sebuah bilangan bulat N ($3 \leq N \leq 100000$).
- Baris kedua berisi N bilangan bulat unik (1 - 1000000) yang merepresentasikan L' .

Format Keluaran

Banyak elemen yang mungkin bisa dipilih menjadi pivot.

Contoh

No	Masukan	Keluaran
1.	5 2 1 3 4 5	3

Penjelasan

$L' = \{2, 1, 3, 4, 5\}$, maka 3 elemen: {3, 4, 5} dapat dipilih menjadi pivot.

Misal, pivot = 3, ini mungkin,karena {2, 1} di kiri pivot lebih kecil dari pivot dan {4, 5} di kanan pivot lebih besar dari pivot.

Namun, pivot = 1, ini tidak mungkin, karena {2} di kiri pivot lebih besar dari pivot.

Catatan

- Keluaran diakhiri dengan newline (\n)
- Download dan gunakan file [boolean.h](#), [listdin.h](#) & [listdin.c](#) yang telah dibuat sebelumnya
- Download dan kumpulkan jawaban dalam file bernama [pivot.c](#)

C

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 1.61 MB

No	Score	Verdict	Description
2	10	Accepted	0.00 sec, 1.51 MB
3	10	Accepted	0.00 sec, 1.67 MB
4	10	Accepted	0.00 sec, 1.72 MB
5	10	Accepted	0.00 sec, 1.60 MB
6	10	Accepted	0.00 sec, 1.51 MB
7	10	Accepted	0.01 sec, 1.72 MB
8	10	Accepted	0.00 sec, 1.71 MB
9	10	Accepted	0.06 sec, 2.21 MB
10	10	Accepted	0.01 sec, 2.64 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Intervalay

Deskripsi

List dapat digunakan untuk menyimpan interval-interval $\langle start, stop \rangle$. Menggunakan *zero-based indexing*, indeks genap digunakan untuk menyimpan nilai *start* dan indeks ganjil digunakan untuk menyimpan nilai *stop*. Contohnya yaitu list [1, 3, 2, 4] menyimpan interval $\langle 1, 3 \rangle$ dan $\langle 2, 4 \rangle$.

Dua buah interval dikatakan beririsan jika **setidaknya** salah satu anggota dari interval pertama berada di dalam interval kedua. Misalnya, interval $\langle 1, 3 \rangle$ dan $\langle 2, 4 \rangle$ beririsan karena 2 berada di dalam interval $\langle 1, 3 \rangle$.

Diberikan *list* interval yang tidak mempunyai irisan, sebut saja intervalay. Intervalay sudah diurutkan berdasarkan nilai *start*-nya dari interval terkecil ke terbesar. Jika kurang jelas, Anda dapat melihat contoh ini.

- [1, 3, 6, 9], bisa menjadi intervalay
- [1, 2, 2, 5], tidak mungkin menjadi intervalay karena $\langle 1, 2 \rangle$ beririsan dengan $\langle 2, 5 \rangle$
- [4, 5, 1, 2], tidak mungkin menjadi intervalay karena tidak terurut berdasarkan *start*-nya secara membesar. Seharusnya $\langle 1, 2 \rangle$ disimpan sebelum $\langle 4, 5 \rangle$.

Ingin dimasukkan sebuah interval baru, sebut saja newInterval. Masukkanlah newInterval ke dalam intervalay sehingga intervalay tetap tidak mempunyai irisan dan tetap terurut berdasarkan nilai *start* dari interval terkecil ke terbesar.

Format Masukan

- Baris pertama adalah panjang *list* intervalay, sebut saja n. Berarti terdapat n/2 interval dalam *list* intervalay. n adalah bilangan genap.
- Baris kedua berisi n buah bilangan bulat yang merupakan intervalay. Interval-interval yang dimasukkan pasti valid.
- Baris ketiga berisi dua buah bilangan bulat yang merupakan newInterval. newInterval pasti merupakan interval yang valid.

Format Keluaran

Keluaran berisi intervalay yang sudah dimasukkan newInterval, dipisahkan spasi dan diakhiri newline (\n).

Contoh

Masukan	Keluaran	Penjelasan
4 1 3 6 9 2 5	1 5 6 9	intervalay = [$\langle 1, 3 \rangle$, $\langle 6, 9 \rangle$] newInterval = $\langle 2, 5 \rangle$ $\langle 1, 3 \rangle$ dari intervalay beririsan dengan $\langle 2, 5 \rangle$ (newInterval), sehingga di-merge dan menghasilkan $\langle 1, 5 \rangle$. Hasilnya adalah [1, 5, 6, 9]
10 1 2 3 5 6 7 8 10 12 16 4 8	1 2 3 10 12 16	intervalay = [$\langle 1, 2 \rangle$, $\langle 3, 5 \rangle$, $\langle 6, 7 \rangle$, $\langle 8, 10 \rangle$, $\langle 12, 16 \rangle$] newInterval = $\langle 4, 8 \rangle$ $\langle 3, 5 \rangle$, $\langle 6, 7 \rangle$, dan $\langle 8, 10 \rangle$ dari intervalay beririsan dengan $\langle 4, 8 \rangle$ (newInterval), sehingga di-merge dan menghasilkan $\langle 3, 10 \rangle$. Hasilnya adalah [1, 2, 3, 10, 12, 16]
0 2 5	2 5	intervalay = [] newInterval = $\langle 2, 5 \rangle$ Hasilnya adalah [2, 5]

Petunjuk

- Anda dapat membuat *list* dinamis baru untuk menyimpan hasilnya, gunakan fungsi insertLast untuk memasukkan angka baru
- Lakukan *merge* pada interval yang beririsan

Catatan

- Jangan lupa mendealokasikan memori yang sudah dialokasikan!
- Tulis jawaban dalam file **intervalay.c**

C

Score: 100

Blackbox

Score: 100

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, 1.60 MB
5	10	Accepted	0.00 sec, 1.65 MB
6	10	Accepted	0.00 sec, 1.64 MB
7	10	Accepted	0.00 sec, 1.55 MB
8	10	Accepted	0.00 sec, 1.59 MB
9	10	Accepted	0.00 sec, 1.64 MB
10	10	Accepted	0.00 sec, 1.59 MB

Jump to...