

Question **1**

Tries remaining:
10

Marked out of
100.00

Time limit	1.5 s
Memory limit	64 MB

Deque atau **Double Ended Queue** adalah versi umum dari struktur data **Queue** yang memungkinkan penyisipan dan penghapusan di kedua ujungnya.

Buatlah sebuah program **Main.java** (Buatlah sendiri) yang menggunakan **Deque** dari **Collection Java** untuk memenuhi spesifikasi program sebagai berikut:

- Program menerima input sebagai berikut:
 - Pada baris pertama, berisi bilangan bulat positif **Q** ($1 \leq Q \leq 100000$) menyatakan banyaknya operasi yang akan dilakukan
 - **Q** baris berikutnya berisi salah satu dari operasi berikut
 - **addFirst X** → Masukkan bilangan bulat **X** kedalam **Deque** dari depan
 - **addLast X** → Masukkan bilangan bulat **X** kedalam **Deque** dari belakang
 - **pollFirst** → Mengeluarkan nilai bilangan di indeks pertama **Deque** ke layar, kemudian menghapus bilangan tersebut
 - **pollLast** → Mengeluarkan nilai bilangan di indeks terakhir **Deque** ke layar, kemudian menghapus bilangan tersebut
 - **reverse** → Melakukan reverse semua elemen pada **Deque**
- Program mengeluarkan output sebagai berikut
 - Untuk setiap operasi **pollFirst** dan **pollLast** keluarkan bilangan yang sesuai
 - apabila pada operasi tersebut **Deque** sedang kosong, maka keluarkan **“EMPTY”** (tanpa tanda petik) pada layar

Contoh Input

```
15
addFirst 93
pollFirst
pollLast
reverse
pollLast
reverse
addFirst 80
addLast 53
addLast 50
pollFirst
pollFirst
reverse
addLast 66
reverse
pollFirst
```

Contoh Output

```
93
EMPTY
EMPTY
80
53
66
```

Submit file **Main.java**.

Hints:

- Anda dapat menggunakan **Deque** sebagai berikut, **Deque<Integer> deque = new ArrayDeque<Integer>()**
- Untuk melakukan operasi yang dibutuhkan, gunakan *method* yang ada pada format input (kecuali **reverse** yang bukan merupakan *method Deque*)

Java 8

Maximum size for new files: 512MB, maximum attachments: 1

You can drag and drop files here to add them.

Run

Check

Question **2**

Tries remaining:
10

Marked out of
100.00

Time limit	1 s
Memory limit	64 MB

Buatlah sebuah program Java yang menerima barisan bilangan bulat yang dipisahkan oleh spasi kemudian mengembalikan list baru yang terurut secara menaik tanpa ada nilai yang duplikat. Jangan lupa new line pada akhir baris

Aturan masukan yang diterima:

- List input dapat mengandung bilangan negatif
- List input dapat berupa list kosong
- List input dapat mengandung hingga 1000 bilangan

Asumsi:

- List input pasti berupa list bilangan bulat yang dipisahkan oleh spasi (contoh: 5 3 9 1 3 8 1 2 2)

Contoh:

Input: 5 3 9 1 3 8 1 2 2

Output: [1, 2, 3, 5, 8, 9]


Silahkan lengkapi file [Main.java](#)

Hint:

- Gunakan konsep collection List dan Set
- Gunakan fungsi .split untuk memisahkan elemen dalam string
- Format output adalah format print ArrayList. Silahkan langsung print variabel bertipe ArrayList

Java 8

Maximum size for new files: 512MB, maximum attachments: 1



[Files](#)

You can drag and drop files here to add them.

Run

Check

Time limit	1 s
Memory limit	64 MB

Student Sorter

Buatlah sebuah program Java yang mengelola informasi mengenai siswa dan nilai-nilai mereka dalam beberapa mata pelajaran. Program ini akan menghitung nilai rata-rata (GPA) dari setiap siswa dan mengurutkan mereka berdasarkan GPA.

Program harus memenuhi persyaratan berikut:

Dalam kelas StudentSorter:

- Terdapat kelas Student dengan atribut name (String) dan courseGrades (Map<String, Integer>). Kelas Student memiliki method `addCourseGrade` untuk menambahkan nilai mata pelajaran dan `getGPA` menghitung GPA.
- Terdapat kelas StudentComparator yang mengimplementasikan Comparator. Kelas ini harus mengurutkan objek Student berdasarkan GPA. Kelas ini meng-override method `compare` yang melakukan komparasi siswa s1 dan s2. Cara kerja komparasi adalah jika $s1 < s2$ method ini akan mengembalikan bilangan negatif, nol jika sama dan positif jika $s1 > s2$. Bilangan tersebut bebas.
- Terdapat static method `sortStudentsByGPA` yang mengembalikan list objek Student yang sudah diurutkan dengan StudentComparator

Lengkapilah file `StudentSorter.java`.

Submit kembali file `StudentSorter.java` yang telah berisi jawaban Anda.

Java 8

Maximum size for new files: 512MB, maximum attachments: 1

[Files](#)

You can drag and drop files here to add them.

Run

Check