

---

**Started on** Sunday, 23 March 2025, 1:58 PM

---

**State** Finished

---

**Completed on** Sunday, 23 March 2025, 2:16 PM

---

**Time taken** 17 mins 42 secs

---

**Grade** 300.00 out of 300.00 (100%)

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

# Magic Calculator

Anda adalah seorang unpaid intern di suatu perusahaan dan atasan anda menyuruh anda untuk membuat suatu kalkulator ajaib. Kalkulator ini dapat memberi tahu pengguna tipe, nilai maksimum dan nilai minimum untuk setiap angka yang dimasukkan ke dalam kalkulator tersebut.

Tugas anda adalah melengkapi file `MagicCalculator.java`

## Format Masukan

Metode `calculate` akan menerima array of `Numbers` yang dapat bertipe `Byte`, `Integer`, `Double`, `Float`, `Short`, `Long`

## Format Keluaran

Baris pertama akan mencetak string "=== MULAI KALKULASI ===" kemudian setiap angka akan melakukan aksi ini : Pertama, program akan mencetak tipe angka tersebut.

- Integer : "<angka> adalah tipe integer"
- Byte : "<angka> adalah tipe byte"
- Double : "<angka> adalah tipe double"
- Float : "<angka> adalah tipe float"
- Short : "<angka> adalah tipe short"
- Long : "<angka> adalah tipe long" Kemudian, program akan mencetak nilai maksimum angka tersebut. diikuti dengan mencetak nilai minimum angka tersebut.

KHUSUS UNTUK TIPE `Float` dan `Double` : Program akan mencetak `true` apabila nilai angka tersebut infinite dan `false` bila bukan.

Baris terakhir akan mencetak string "=== KALKULASI SELESAI ==="

## Contoh Keluaran 1

```
=== MULAI KALKULASI ===
10 adalah tipe integer
2147483647
-2147483648
20.5 adalah tipe double
1.7976931348623157E308
4.9E-324
false
3 adalah tipe byte
127
-128
=== KALKULASI SELESAI ===
```

Hint :

1. Gunakan `MAX_VALUE` dan `MIN_VALUE` untuk soal ini!

attachment : `MagicCalculator.java`

Submit file [MagicCalculator.java](#)

Java 8

 [MagicCalculator.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.06 sec, 28.95 MB
2	10	Accepted	0.06 sec, 28.39 MB
3	10	Accepted	0.06 sec, 28.99 MB
4	10	Accepted	0.06 sec, 28.99 MB
5	10	Accepted	0.06 sec, 28.93 MB
6	10	Accepted	0.06 sec, 30.77 MB
7	10	Accepted	0.06 sec, 28.33 MB
8	10	Accepted	0.06 sec, 29.86 MB
9	10	Accepted	0.06 sec, 27.87 MB
10	10	Accepted	0.07 sec, 28.67 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Anda diberikan data PII (Personally Identifiable Information) yang berupa string email address atau phone number Tugas anda disini adalah mengamankan data PII tersebut dengan melakukan masking terhadapnya.

Definisi Email address:

- Terdiri atas dua bagian yaitu **name** dan **domain** yang dipisahkan dengan karakter **@**
- Karakter-karakter pada **name** dan **domain** berupa alphanumeric (huruf kapital, kecil, dan angka)

Prosedur Masking Email Address:

- Semua huruf pada **name** dan **domain** dikonversi menjadi huruf kecil
- Selain karakter pertama dan terakhir pada **name**, semua karakter ditengah dihapus kemudian ditambah dengan **\*\*\*\*\*** (5 asterisk)

Contoh: **eppy@asistenbaikkatanya123.com**

Hasil setelah masking: **e\*\*\*\*\*y@asistenbaikkatanya123.com**

Definisi Phone Number:

- Terdiri atas 10-13 digit
- 10 digit terakhir merupakan **local number**
- 0-3 digit pertama merupakan **country code**
- Digit-digitnya dapat dipisahkan dengan karakter pemisah '+', '-', '(', ')', ' ', ''

Prosedur Masking Phone Number:

- Hapus semua karakter pemisah
- Mask nomor telepon dalam format:
  - **\*\*\*-\*\*\*-XXXX** jika **country code** memiliki 0 digit
  - **+\*-\*\*\*-\*\*\*-XXXX** jika **country code** memiliki 1 digit
  - **+\*\*-\*\*\*-\*\*\*-XXXX** jika **country code** memiliki 2 digit
  - **+\*\*\*-\*\*\*-\*\*\*-XXXX** jika **country code** memiliki 3 digit
- **XXXX** merupakan 4 digit terakhir dari **local number**

Contoh: **+1(324)567-8902** Hasil setelah masking: **+\*-\*\*\*-\*\*\*-8902**

Lengkapilah metode-metode pada file [MaskPII.java](#). Kemudian, Submit file **MaskPII.java**

Java 8

 [MaskPII.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.31 sec, 28.43 MB
2	10	Accepted	0.35 sec, 27.82 MB
3	20	Accepted	0.22 sec, 27.83 MB
4	20	Accepted	0.29 sec, 29.64 MB

No	Score	Verdict	Description
5	20	Accepted	0.11 sec, 27.89 MB
6	20	Accepted	0.12 sec, 29.73 MB

Question **3**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Buatlah sebuah program yang dapat memproses sebuah string. Lengkapi file [`WordProcessor.java`](#).

Submit file **`WordProcessor.java`**.

Java 8

 [WordProcessor.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	25	Accepted	0.36 sec, 30.40 MB
2	25	Accepted	0.38 sec, 27.90 MB
3	25	Accepted	0.32 sec, 30.13 MB
4	25	Accepted	0.39 sec, 28.19 MB

[◀ Remedial Praktikum 2](#)

Jump to...

[Praktikum 3 ▶](#)