

Started on	Sunday, 5 March 2023, 4:26 PM
State	Finished
Completed on	Wednesday, 8 March 2023, 9:52 PM
Time taken	3 days 5 hours
Grade	600.00 out of 600.00 (100%)

Question **1**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

## Numbers - Minimum Operations

Buatlah sebuah program yang dapat mengembalikan sebuah integer yang merupakan operasi minimum untuk mencapai array yang diinginkan. Parameter arrNum pada fungsi `minimumOperationsNeeded` merupakan array yang diinginkan. Pada awalnya setiap elemen pada sebuah array terdiri dari semua elemen dengan nilai 0, lalu Anda dapat melakukan 2 operasi, yaitu operasi pertama yang dapat menambah satu elemen dengan 1 atau operasi kedua yang dapat mengalikan setiap elemen dengan 2 untuk mencapai array yang diinginkan. Sebagai contoh, apabila arrNum berisi sebuah elemen [2,2], maka diperlukan minimum 3 operasi yaitu dengan operasi pertama sebanyak dua kali dengan menambahkan 1 pada elemen pertama, [0,0] -> [1,0] lalu menambahkan 1 pada elemen kedua, [1,0] -> [1,1] selanjutnya dengan menggunakan operasi 2 sehingga [1,1] -> [2,2].

Lengkapi program [MinOp.java](#) lalu Submit file tersebut

\*Hint, parameter pada fungsi `minimumOperationsNeeded` merupakan array of Numbers, sehingga perlu diubah terlebih dahulu ke dalam tipe integers, caranya adalah dengan menggunakan method `intValue()` untuk setiap elemen pada array input tersebut.

Hint kedua, idenya dapat dimulai dari target lalu menuju ke state awal (setiap elemen bernilai 0) jika semuanya genap maka dapat langsung dibagi 2, namun jika masih terdaapt elemen ganjil maka dapat dikurangi terlebih dahulu dengan 1 untuk setiap elemen yang bernilai ganjil

Java 8

 [MinOp.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.07 sec, 27.84 MB
2	20	Accepted	0.07 sec, 28.45 MB
3	20	Accepted	0.10 sec, 28.13 MB
4	20	Accepted	0.08 sec, 28.42 MB
5	20	Accepted	0.07 sec, 28.03 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

# Strings - WordProcessor

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	0.08 sec, 28.73 MB
2	25	Accepted	0.08 sec, 28.98 MB
3	25	Accepted	0.08 sec, 29.15 MB
4	25	Accepted	0.07 sec, 26.64 MB

Question **3**

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) dan titik (hanya untuk **domain**)

**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: **Kadek@Mars1729.com**

Hasil setelah masking: **k\*\*\*\*\*k@mars1729.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 phone number dengan 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	16	Accepted	0.07 sec, 28.94 MB
2	16	Accepted	0.07 sec, 28.83 MB
3	16	Accepted	0.07 sec, 29.07 MB
4	16	Accepted	0.07 sec, 28.19 MB
5	16	Accepted	0.07 sec, 28.04 MB
6	20	Accepted	0.07 sec, 27.76 MB

Question **4**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Diberikan suatu array statik bertipe **Number**. Anda ditugaskan untuk menghitung penjumlahan elemen-elemen array dengan tipe data yang sama

Tipe data yang mungkin terdapat pada array tersebut: **Byte**, **Integer**, **Double**, **Float**, **Short**, **Long**

**Contoh:**

Array masukan: [2L, 4.5f, 3.5, 45L, 100, 5, 2] // [Long, Float, Double, Long, Short, Byte, Byte]

Array keluaran: [47L, 4.5f, 3.5, 100, 7, null, null] // [Long, Float, Double, Short, Byte, null, null]

**Penjelasan:**

Terdapat 2 elemen dengan tipe data Long, 1 Float, 1 Double, 1 Short dan 2 Byte

Sehingga, jika elemen dengan tipe data yang sama dijumlahkan:

[(2L + 45L), (4.5f), (3.5), (100), (5+2)]

[47L, 4.5f, 3.5, 100, 7, null, null]

Karena tersisa dua tempat kosong pada array keluaran, maka keduanya diisi dengan null

**Perhatikan :**

- Urutan tipe data array keluaran haruslah sama dengan array masukan.
- Panjang array keluaran haruslah sama dengan array masukan. Jika terdapat tempat kosong setelah dilakukan penjumlahan, tempat tersebut diisi dengan null

**Hint:**

- Untuk inialisasi array statik bisa dengan: **Number[] answer = new Number[arrLen]** dimana arrLen merupakan panjang array
- Untuk mengecek apakah dua objek merupakan kelas yang sama bisa dengan: **obj1.getClass().equal(obj2.getClass())**
- Untuk mengecek apakah suatu kelas merupakan subclass dari kelas lain bisa dengan keyword: **instanceof**

Lengkapilah metode **numberSum** pada file [NumberSum.java](#). Kemudian, Submit file **NumberSum.java**

Java 8

⬇

 [NumberSum.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.07 sec, 28.03 MB
2	10	Accepted	0.07 sec, 28.53 MB
3	10	Accepted	0.07 sec, 27.94 MB
4	10	Accepted	0.08 sec, 28.74 MB
5	10	Accepted	0.08 sec, 29.12 MB
6	10	Accepted	0.07 sec, 27.94 MB
7	10	Accepted	0.07 sec, 27.76 MB
8	10	Accepted	0.08 sec, 28.51 MB
9	10	Accepted	0.07 sec, 27.94 MB
10	10	Accepted	0.07 sec, 27.89 MB

Question **5**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

# Strings - Password Manager

Buatlah sebuah program yang dapat mengatur password. Lengkapi file [PasswordManager.java](#).

Submit file **PasswordManager.java**.

Java 8

⬇

 [PasswordManager.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	25	Accepted	0.24 sec, 28.92 MB
2	25	Accepted	0.38 sec, 29.78 MB
3	50	Accepted	0.92 sec, 28.06 MB

Question **6**

Correct

Mark 100.00 out of 100.00


Time limit	1 s
Memory limit	64 MB

# Strings - PalindromeProcessor

Buatlah sebuah program yang dapat mengembalikan substring palindrom terpanjang. Lengkapi file [PalindromeProcessor.java](#). Buat yang belum tahu mengenai substring, substring merupakan suatu sekuens kontigu yang terdiri atas karakter didalam sebuah string. Sebagai contoh, misal terdapat suatu string 'apple', maka semua substring yang mungkin adalah : "a", "ap", "app", "appl", "apple", "p", "pp", "ppl", "pple", "pl", "ple", "l", "le" "e", "". Pada soal kali ini, anda diminta untuk mengembalikan sebuah substring yang berupa palindrome dari sebuah string, namun substring tersebut harus merupakan substring terpanjang dari semua kemungkinan substring yang ada. Misal terdapat sebuah string "qweewqx", "ee" dan "weew" merupakan sebuah substring yang berupa palindrome, namun bukan yang terpanjang, sehingga yang harus dikembalikan merupakan substring "qweewq".

Submit file **PalindromeProcessor.java**.

Java 8 

 [PalindromeProcessor.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	20	Accepted	0.15 sec, 27.84 MB
2	20	Accepted	0.10 sec, 28.32 MB
3	20	Accepted	0.11 sec, 29.86 MB
4	20	Accepted	0.11 sec, 28.32 MB
5	20	Accepted	0.13 sec, 28.95 MB