

Started on	Tuesday, 17 June 2025, 7:36 PM
State	Finished
Completed on	Tuesday, 17 June 2025, 7:50 PM
Time taken	14 mins 13 secs
Grade	300.00 out of 300.00 (100%)

Time limit	1 s
Memory limit	64 MB

# Techbrosgram

Kalian adalah seorang mahasiswa STI yang memiliki *founder mindset*: "My body is a machine that turns stress to B2C SaaS". Suatu hari, kalian memiliki ide untuk menciptakan media sosial baru khusus untuk *techbros*. Pasalnya, kalian terkadang didera masyarakat umum di media sosial lain karena tidak napah tanah. Mari wujudkan *startup* kalian tersebut!

## Penjelasan Soal

- Terdapat kelas **Techbrosgram** yang menyimpan atribut **bros** berisi kumpulan Bro, **posts** berisi kumpulan Post, dan **postCounter** yang diinkremen sebagai ID untuk tiap Post
- Metode **registerBros** akan menerima sebuah nama, membuat Bro baru, dan menyimpannya ke dalam **bros** di atribut Techbrosgram.
- Metode **createPost** akan menerima nama Bro pembuat Post, isi Post, membuat Post, lalu menyimpannya ke **posts** di atribut Techbrosgram. ID Post dibuat berdasarkan **postCounter**. Nama Bro tidak dijamin valid.
- Metode **likePost** akan menerima nama Bro yang memberi *like* dan ID Post yang dilike. Sebuah Post tidak dapat di-*like* oleh pembuatnya sendiri. Nama Bro dan ID Post tidak dijamin valid. Catat nama Bro yang memberi *like* di dalam set pada objek Post terkait.
- Kelas *inner* **Bro** menyimpan nama Bro dan kumpulan Post yang pernah dibuat.
- Kelas *inner* **Post** menyimpan ID (berasal dari postCounter), konten, penulis (suatu Bro), dan daftar Bro yang menyukai Post tersebut.


**PERHATIKAN *EDGE CASE* YANG MUNGKIN MUNCUL. BACA PETUNJUK DI KOMENTAR DENGAN SAKSAMA.**

## File

- Lengkapilah file [Techbrosgram.java](#)

Submit kembali file **Techbrosgram.java** yang telah berisi jawaban Anda.

Java 8 ▾

 [Techbrosgram.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	5	Accepted	0.15 sec, 33.69 MB
2	5	Accepted	0.15 sec, 33.35 MB
3	5	Accepted	0.15 sec, 33.23 MB
4	5	Accepted	0.15 sec, 33.73 MB
5	5	Accepted	0.15 sec, 32.14 MB
6	5	Accepted	0.15 sec, 33.40 MB
7	5	Accepted	0.15 sec, 33.80 MB

No	Score	Verdict	Description
8	5	Accepted	0.15 sec, 33.46 MB
9	5	Accepted	0.15 sec, 31.77 MB
10	5	Accepted	0.15 sec, 33.70 MB
11	5	Accepted	0.16 sec, 32.18 MB
12	5	Accepted	0.15 sec, 33.36 MB
13	5	Accepted	0.15 sec, 33.79 MB
14	5	Accepted	0.15 sec, 33.34 MB
15	5	Accepted	0.16 sec, 33.80 MB
16	5	Accepted	0.16 sec, 33.52 MB
17	5	Accepted	0.15 sec, 32.11 MB
18	5	Accepted	0.15 sec, 33.53 MB
19	5	Accepted	0.16 sec, 33.46 MB
20	5	Accepted	0.16 sec, 33.44 MB

Time limit	1 s
Memory limit	64 MB

# ART PRINT

Seorang teman kalian sedang membuka booth di suatu art market bernama CVXX. Kalian ditugaskan untuk menjaga booth teman kalian yang menjual art print.

Art print ini ditumpuk dalam suatu **stack**. Pertama, program akan meminta berapa perintah yang akan dijalankan. Kemudian program akan menjalankan beberapa perintah berikut :

"tambah" -> menambahkan art print ke dalam stack art print. Akan menerima SATU KATA yang merupakan nama karakter setelahnya. Setelah berhasil menambahkan, akan mencetak **sudah menambahkan <nama karakter>**. UNTUK SOAL INI, ASUMSIKAN INPUT NAMA KARAKTER SELALU BENAR!

Contoh penggunaan :

```
tambah gojo
sudah menambahkan gojo
```

"beli" -> mengambil art print dari stack art print. Apabila stock habis akan mencetak **maaf stock habis** namun apabila stock masih ada akan mencetak **telah membeli <nama karakter>** dengan karakter yang diambil merupakan karakter yang ada di paling atas stack

Contoh penggunaan :

```
beli
telah membeli gojo
```

"lihat" -> Memperlihatkan art print yang ada di paling atas stack. Akan mencetak **print paling atas adalah <nama karakter>** namun apabila stack kosong maka akan mencetak **maaf stock habis**

Contoh penggunaan :

```
lihat
print paling atas adalah gojo
```

"habis?" -> Cek apakah stack art print sudah habis atau belum. Apabila stock belum habis akan mencetak **masih ada kok** dan apabila stock sudah habis akan mencetak **ya, habis**

Contoh peggunaan :

```
habis?
ya, habis
```

# GUIDE PERINTAH

Perintah	Input	Output
tambah <nama karakter>	tambah	Pikachusudah menambahkan Pikachu
beli	beli	telah membeli <nama karakter>
beli saat stack kosong	beli	maaf stock habis
lihat	lihat	print paling atas adalah <nama karakter>
lihat saat stack kosong	lihat	maaf stock habis
habis? saat stack kosong	habis?	ya, habis
habis? saat stack tidak kosong	habis?	masih ada kok

# CONTOH PROGRAM

```
4
tambah gojo
sudah menambahkan gojo
lihat
print paling atas adalah gojo
beli
telah membeli gojo
habis?
ya, habis
```

# HINT

- gunakan `Stack<String> stack = new Stack<>();` untuk inisiliasi stack
- untuk memudahkan, import `java.util.*`
- gunakan fungsi stack sebagai berikut : `push()`, `pop()`, `empty()` dan `peek()`
- `push(x)` -> push elemen x ke dalam stack
- `pop()` -> pop elemen paling atas stack
- `empty()` -> mengembalikan true stack kosong, false apabila stack berisi
- `peek()` -> mengembalikan elemen paling atas stack
- contoh penggunaan : `stack.push(x)`

Submit Main.java!

Java 8 ▾

 [Main.java](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.07 sec, 28.19 MB
2	10	Accepted	0.07 sec, 29.91 MB
3	10	Accepted	0.07 sec, 26.91 MB
4	10	Accepted	0.08 sec, 28.76 MB
5	10	Accepted	0.07 sec, 28.97 MB
6	10	Accepted	0.07 sec, 28.34 MB
7	10	Accepted	0.07 sec, 27.80 MB
8	10	Accepted	0.07 sec, 28.19 MB
9	10	Accepted	0.07 sec, 28.34 MB
10	10	Accepted	0.07 sec, 28.30 MB

Time limit	1 s
Memory limit	64 MB

## Konvoi

Persil, sebuah klub sepak bola yang berasal dari Spakbor Hills, baru saja meraih puncak klasemen pada liga yang diikuti. Untuk merayakannya, mereka berencana untuk mengadakan konvoi yang akan diramaikan oleh pendukungnya.

Sebagai bentuk upaya menjaga ketertiban, Ketua Pendukung Persil Jaya meminta kamu untuk membantu mereka dalam melakukan pendataan terhadap siapa saja yang mengikuti konvoi tersebut. Ada 4 kelas yang harus kamu implementasikan:

- 1. Kendaraan (**Kendaraan.java**)
- 2. RuteKonvoi (**RuteKonvoi.java**)
- 3. SegmenRute (**RuteKonvoi.java**)
- 4. KonvoiPersil (**KonvoiPersil.java**)

Attachment: [attachment.zip](#)

**Submit kembali file Kendaraan.java, RuteKonvoi.java, dan KonvoiPersil.java dalam bentuk zip.**

Hint:

**Gunakan metode keySet() dari interface Map untuk mendapatkan Set dari suatu Map.**

Java 8 ▾

 [3.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	6	Accepted	0.06 sec, 28.80 MB
2	6	Accepted	0.06 sec, 30.69 MB
3	6	Accepted	0.06 sec, 28.13 MB
4	6	Accepted	0.06 sec, 28.38 MB
5	6	Accepted	0.06 sec, 28.99 MB
6	6	Accepted	0.06 sec, 28.28 MB
7	6	Accepted	0.06 sec, 26.72 MB
8	6	Accepted	0.06 sec, 27.89 MB
9	6	Accepted	0.06 sec, 28.98 MB
10	6	Accepted	0.06 sec, 30.04 MB
11	6	Accepted	0.06 sec, 28.85 MB

No	Score	Verdict	Description
12	6	Accepted	0.06 sec, 28.26 MB
13	6	Accepted	0.06 sec, 30.43 MB
14	6	Accepted	0.06 sec, 29.16 MB
15	6	Accepted	0.06 sec, 30.91 MB
16	10	Accepted	0.06 sec, 28.77 MB

[◀ Feedback Form](#)

Jump to...

⬆

[Pra-Kuis 2 ▶](#)