

Soal yang diberikan kali ini mengajak peserta untuk memikirkan kira-kira ADT apa (dari yang dipelajari di kelas) yang tepat digunakan untuk menyelesaikan permasalahan yang diberikan.

Di awal, peserta akan diberikan permasalahan dan ADT yang harus digunakan. Tujuannya adalah agar peserta mendapatkan “*feel*” tentang apa saja karakteristik permasalahan yang cocok diselesaikan dengan ADT tertentu. Kemudian, peserta akan diberikan permasalahan tanpa spesifikasi ADT dan peserta harus mampu memilih satu atau beberapa ADT yang **cocok** untuk menyelesaikan permasalahan yang diberikan.

### **Problem 1. Sorting Data**

Program menerima input beberapa baris (hingga EOF). Setiap baris berisi informasi tentang produk dengan format <**nomor id, nama, harga satuan**>.

```
123 nutrisari 4000
124 indomie 3500
455 mizone 6000
768 rinso 9000
766 pepsodent 10000
789 dove 8000
```

Program akan menghasilkan output dengan format <**nama, harga satuan**> dan terurut berdasarkan harga dari yang paling mahal hingga yang paling murah. Gunakan **ADT PriorityQueue** dalam hal ini.

```
pepsodent 10000
rinso 9000
dove 8000
mizone 6000
nutrisari 4000
indomie 3500
```

## Problem 2. Reversing Sentence

Buatlah program yang menerima beberapa baris kalimat hingga EOF.

```
saya pergi ke depok  
di depok saya makan belimbing  
saya juga belajar SDA di depok
```

Program akan menampilkan output berupa kalimat-kalimat yang sama dalam urutan kata-kata yang terbalik.

```
depok ke pergi saya  
belimbing makan saya depok di  
depok di SDA belajar juga saya
```

Gunakan **ADT Stack** dalam permasalahan ini! Anda memang bisa langsung melakukan iterasi/loop dari indeks paling akhir (**dan lebih efisien**). Namun, tujuan utama dari soal ini adalah agar Anda mempelajari karakteristik dari **ADT Stack**. Jadi, Anda harus menggunakan **ADT Stack** di permasalahan kali ini.

## Problem 3. Counting Fruits

Program menerima input berupa rekapitulasi pembelian berbagai macam jenis buah dalam waktu tertentu. Jenis buah bisa banyak dan tidak dibatasi. Setiap baris pada input mempunyai format **<nama buah, banyaknya buah>**.

```
Apel 3  
Jeruk 2  
Apel 1  
Belimbing 2  
Jeruk 4  
Mangga 3  
Jambu 2  
Mangga 1  
Mangga 1
```

Kelapa 2

Program akan menampilkan rekapitulasi untuk masing-masing buah seperti:

Apel 4  
Jeruk 6  
Belimbing 2  
Mangga 5  
Jambu 2  
Kelapa 2

Output **tidak perlu terurut** berdasarkan nama buah atau banyaknya buah. Gunakan **ADT Map** dengan implementasi **HashMap** !

#### **Problem 4. Counting Fruits 2**

Modifikasi solusi dari problem 3 sehingga output akan ditampilkan **secara terurut** berdasarkan nama buahnya (secara alfabetis/leksikografis).

Kira-kira ADT apa yang bisa Anda gunakan ?

Contoh output untuk input yang ditampilkan di soal sebelumnya:

Apel 4  
Belimbing 2  
Jambu 2  
Jeruk 6  
Kelapa 2  
Mangga 5

#### **Problem 5. Valid Expression**

Buatlah sebuah program yang menerima sebuah String dengan karakter-karakter hanya terdiri dari '(' dan ')'. Aturan valid atau tidak validnya sebuah ekspresi dapat dilihat pada contoh-contoh di tabel berikut:

()	Valid
(	Tidak valid
)	Tidak valid
()()()	Valid
()	Tidak valid
((()))	Tidak valid

Implementasikan sebuah *method* yang menerima ekspresi tersebut dan mengembalikan **true** jika ekspresi **valid** dan sebaliknya.

Gunakan **ADT Stack** untuk menyelesaikan masalah ini !

### Problem 6. Wordcount Problem

Buatlah program yang menerima beberapa baris kalimat sebagai input hingga EOF. Program kemudian akan menampilkan kemunculan masing-masing kata dari kalimat-kalimat yang dimasukkan.

Format output adalah **<kata, frekuensi kata>** dan kata-kata diurutkan secara alfabetis.

Contoh input:

```
saya dan kamu pergi ke depok
saya dan kamu makan mie kocok
kamu pergi ke depok dan saya makan mie kocok
mie kocok adalah makanan kesukaan saya
```

Contoh output:

```
saya 4
dan 3
kamu 3
pergi 2
ke 2
depok 2
makan 2
```

mie 3  
kocok 3  
adalah 1  
makanan 1  
kesukaan 1

### Problem 7. Purchasing Fruits

Pertama-tama, program akan menerima input sebuah bilangan bulat **M** yang menandakan **berapa banyak informasi buah** yang akan dimasukkan ke dalam program. Setelah itu, program akan meminta masukan informasi **nama buah** beserta **harga satuannya** sebanyak **M**.

Kemudian, program akan menerima input beberapa baris (hingga EOF) yang terdiri dari **nama pembeli**, **nama buah yang dibeli**, dan **banyaknya buah yang dibeli**. Kita asumsikan bahwa nama orang dan nama buah adalah unik.

Contoh Input:

3  
**jeruk 1000**  
**apel 2000**  
**durian 3000**  
Dodi apel 5  
Dodi jeruk 6  
Rudi apel 1  
Rudi apel 2  
Andi jeruk 3  
Budi apel 3  
Andi jeruk 2  
Budi jeruk 1  
Dodi durian 2

Program akan menampilkan informasi untuk setiap nama pembeli berupa **total harga** yang harus dibayar beserta **nama semua buah yang dibeli** lengkap dengan total banyak buah yang dibeli untuk setiap jenis buah.

Contoh Output:

```
Andi 5000.0 (jeruk,5)
Budi 7000.0 (apel,3)(jeruk,1)
Dodi 22000.0 (apel,5)(durian,2)(jeruk,6)
Rudi 6000.0 (apel,3)
```

Nama pembeli terurut secara alfabetis, dan nama buah juga terurut secara alfabetis.