

## Question 1

Not answered

Marked out of 4.00

Buatlah sebuah program yang menerima dua string dan menentukan apakah kedua string tersebut adalah anagram satu sama lain. Dua string dikatakan anagram jika mereka memiliki karakter yang sama dengan frekuensi yang sama, tetapi urutannya bisa berbeda. Program harus mengabaikan spasi dan tidak peka terhadap huruf besar/kecil.

Input:

Sebuah baris teks, [kalimat 1][#][kalimat 2]

Output:

Cetak:

input merupakan anagram.

atau

input bukan merupakan anagram.

Test case:

Input: Listen # Silent

Output: input merupakan anagram.

Input: Hakozaiki # Rikimaru

Output: input bukan merupakan anagram.

[Template](#)

[TestFile](#)

Hint:

Silahkan gunakan java.util.Arrays; dan method-method yang ada dibawah ini

- `replaceAll("\\s+", "")`: Menghapus semua spasi jika ada.
- `toLowerCase()`: Membuat perbandingan tidak peka huruf besar/kecil.
- `Arrays.sort()`: Mengurutkan karakter.
- `Arrays.equals()`: Memeriksa apakah kedua array karakter sama.
- `char[] arr1 = s1.toCharArray();` // Ubah string s1 ke array of karakter

Acuan penilaian:

Poin 4 : Jika luaran program sudah tepat dan coverage dari test case = 100%

Poin 3: Jika luaran program sudah tepat dan coverage dari test case >= 80%

Poin 2: Jika luaran program sudah tepat namun test case tidak tersedia

Poin 1: Jika luaran program tidak tepat dan test case juga belum tersedia dengan benar

Poin 0: Compile Error

## Response history

Step	Time	Action	State	Marks
<a href="#">1</a>	15/04/25, 13:02	Started	Not yet answered	
<b>2</b>	<b>15/04/25, 15:30</b>	<b>Attempt finished</b>	<b>Not answered</b>	

Question **2**

Not answered

Marked out of 4.00

Buatlah sebuah method `innerArrayChecker` yang menerima dua buah array `outer` dan `inner`. method tersebut akan mengembalikan nilai **true** jika semua angka yang ada pada `inner` **muncul pada outer dengan urutan yang sama**.

```
public static boolean innerArrayChecker(int[] outer, int[] inner){}
```

contoh eksekusi method:

- `innerArrayChecker(new int[]{1, 2, 4, 6}, new int[]{2, 4})` → true
- `innerArrayChecker(new int[]{1, 2, 4, 6}, new int[]{2, 3, 4})` → false
- `innerArrayChecker(new int[]{1, 2, 4, 4, 6}, new int[]{2, 4, 6})` → true
- `innerArrayChecker(new int[]{1, 5, 3, 2, 1}, new int[]{3, 2})` → true

Acuan penilaian:

**Poin 4 :** Jika luaran program sudah tepat dan coverage dari test case = 100% pada method `innerArrayChecker`

**Poin 3:** Jika luaran program sudah tepat dan coverage dari test case >= 80% pada method `innerArrayChecker`

**Poin 2:** Jika luaran program sudah tepat namun test case tidak tersedia atau kesalahan minor

**Poin 1:** Jika luaran program tidak tepat dan test case juga belum tersedia dengan benar

**Poin 0:** Compile Error

{1,2}, {1,2,3}

## Response history

Step	Time	Action	State	Marks
1	15/04/25, 13:02	Started	Not yet answered	
2	15/04/25, 15:30	Attempt finished	Not answered	

Question **3**

Not answered

Marked out of 4.00

```
public class MobilBukanOO {  
    public static void main(String[] args) {  
        String carBrand = "Camry";  
        String carModel = "Sedan";  
        int currentSpeed = 0;  
        currentSpeed = accelerate(currentSpeed, 100);  
        System.out.println(currentSpeed);  
        currentSpeed = brake(currentSpeed, 10);  
        System.out.println(currentSpeed);  
        displayCarInfo(carBrand, carModel);  
    }  
    public static int accelerate(int carSpeed, int increase) {  
        return carSpeed + increase;  
    }  
    public static int brake(int carSpeed, int decrease) {  
        return carSpeed-decrease;  
    }  
    public static void displayCarInfo(String brand, String model) {  
        System.out.println("Car: " + Brand + " " + Model);  
    }  
}
```

Code di atas masih dalam bentuk prosedural, di mana code ini menunjukkan *state* dan *behaviour* dari sebuah obyek Car. Lakukan proses refactor dari code di atas sehingga mengikuti aturan *object oriented* yang tepat. Anda bisa menggunakan template yang disediakan. Anda diwajibkan membuat test case dan pastikan *coverage*  $\geq 85\%$ . Luaran yang diharapkan ketika class IniMobilOO (IniMobilOO.java) dijalankan:

```
0  
150  
125  
Car: Camry Sedan
```

Berkas template: [Template](#)

Acuan penilaian:

**Poin 4** : Jika luaran class IniMobilOO (IniMobilOO.java) sudah tepat dan coverage dari test case  $\geq 85\%$

**Poin 3**: Jika luaran class IniMobilOO (IniMobilOO.java) sudah tepat namun coverage dari test case  $< 85\%$

**Poin 2**: Jika luaran class IniMobilOO (IniMobilOO.java) sudah tepat namun test case tidak tersedia

**Poin 1**: Jika luaran class IniMobilOO (IniMobilOO.java) tidak tepat dan test case juga belum tersedia dengan benar

## Response history

Step	Time	Action	State	Marks
<a href="#">1</a>	15/04/25, 13:02	Started	Not yet answered	
<b>2</b>	<b>15/04/25, 15:30</b>	<b>Attempt finished</b>	<b>Not answered</b>	

## Question 4

Not answered

Marked out of 4.00

Pada [Tugas Pemrograman 2](#), anda belum diperkenankan menggunakan [inheritance](#). Namun terlihat sekali sebenarnya ada beberapa hal yang bisa diperbaiki dengan menerapkan [inheritance](#).

Pada soal kali ini, coba anda perhatikan, ada beberapa method **login(username,password)** yang sama pada beberapa class dalam class diagram TP2 tersebut. Hanya memperhatikan isu tersebut, implementasikan ulang atau perbaiki atau tambahkan, *class-class* yang terkait dengan menerapkan [inheritance](#). Silahkan anda memperbaiki beberapa hal lain terkait [inheritance](#) yang bisa diperbaiki sekalian. Perhatikan waktu dalam pengerjaannya.

Anda dapat meng-**copy-paste** dari submission TP2 anda dan lakukan perubahan secara minimal.

Yang harus anda lakukan dan submit:

- Perubahan code agar menerapkan [inheritance](#) untukantisipasi permasalahan tersebut diatas.  
Sebuah berkas Readme.md berisi secara singkat tapi jelas:
  - Berkas-berkas apa saja yang diubah
  - Perubahan apa yang dilakukan permasing-masing berkas
- Compress dengan format .zip, dengan penamaan: ProgrammingInheritance\_NPM\_namaSingkat.zip
- Submit pada tempat yang disediakan.

Acuan Penilaian:

- Nilai 4: Penjelasan pada Readme.md singkat dan jelas; dan sempurna penerapan [inheritance](#); tidak ada duplikasi; dan Unit test berjalan baik.
- Nilai 3: Penjelasan pada Readme.md singkat dan jelas, dan program berjalan, tapi penerapan tidak sempurna, atau masih ada duplikasi atau unit test tidak berjalan baik.
- Nilai 2: Readme.md kurang jelas. Atau ada upaya menerapkan [inheritance](#) yang benar, namun belum selesai atau program tidak bisa dicompile.
- Nilai 1: Tidak ada Readme.md. Atau Ada perubahan tapi tidak significant, masih terlalu sedikit perubahannya, atau terlalu jauh dari jawaban yang diharapkan (misalnya masih kurang dari 50%)

## Response history

Step	Time	Action	State	Marks
<a href="#">1</a>	15/04/25, 13:02	Started	Not yet answered	
<b>2</b>	<b>15/04/25, 15:30</b>	<b>Attempt finished</b>	<b>Not answered</b>	