Problem 1(OOP)

Intuition:

When a method in a subclass overrides a method in superclass, it is still possible to call the overridden method using **super** keyword. If you write *super.func()* to call the function *func()*, it will call the method that was defined in the superclass.

You are given a partially completed code in the editor. Modify the code so that the code prints the following text:

Hello I am a motorcycle, I am a cycle with an engine.
My ancestor is a cycle who is a vehicle with pedals.

Solution:

```
import java.util.*;
import java.io.*;

class BiCycle{
    String define_me(){
        return "a vehicle with pedals.";
    }
}

class MotorCycle extends BiCycle{
    String define_me(){
        return "a cycle with an engine.";
    }

    MotorCycle(){
        System.out.println("Hello I am a motorcycle, I am "+ define me());

        String temp=super.define me(); //Fix this line

        System.out.println("My ancestor is a cycle who is "+ temp );
    }

}
class Solution{
    public static void main(String []args){
        MotorCycle M=new MotorCycle();
    }
}
```

Problem 2(DBMS)

Intuition:

Solution:

```
# Write your MySQL query statement below
with nun_tv_15 as (
    select tiv_2015
    from insurance
    group by tiv_2015
    having count(*) > 1
), un_ll as(
    select lat, lon
    from insurance
    group by lat, lon
    having count(*) = 1
)
select round(sum(tiv_2016),2) as tiv_2016
from insurance i
where tiv_2015 in (select tiv_2015 from nun_tv_15)
and (lat, lon) in (select lat, lon from un_ll);
```