**Category**: Conditional Statements

1. Write a method that given an integer num, it performs the following conditional actions:

* Print Tom if num is odd
* Print Jerry, if  num is even and > 2 but < 9
* Print Tom, if  num >=  6 but <= 20,
* print Jerry, if num is even and > 25

Use the following stub code.

import java.io.\*;

import java.util.\*;

import java.text.\*;

import java.math.\*;

import java.util.regex.\*;

public class ConditionalSolution {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

String ans="";

//Complete the code

System.out.println(ans);

}

}

**Category:** Object Sorting

Create a Student class that represents the following information of a student: id, name, and age.

Create a StudentSorter class with a sortStudent() method that accepts Student object. Sort the list based on their age in decreasing order. For student having same age, sort based on their name. For students having same name and age, sort them according to their iD.

**Category**: Comparator

Consider the following Pokemon class:

class Pokemon{

String name;

int power;

Pokemon(String name, int power){

this.name = name;

this.power = power;

}

}

Create a PokemonComparator class with a comparePokemon() method that accepts an array of Pokemon objects. The comparator should sorts the pokemons in order of decreasing score. If two pokemons have the same score, sort them alphabetically by name.

You need to use the Comparator class in your implementation.

**Category**: Collections

Write a method that accepts a Map object having two key-value pairs with the keys val1 and val2. Modify and return the given map as follows:

* If the key "val1" has a value, set the key "val2" to have that value, and
* Set the key "val1" to have the value "" (empty string).

For example 1: The map {"val1": "java", "val2": "c++"} should return {"val1": "", "val2": "java"}

For example 2: The map {"val1": "mars", "val2": "saturn"} should return {"val1": "", "val2": "mars"}

**Stub**

public class MapSolution {

public Map<String, String> mapTest(Map<String, String> map)

{

/\* Complete the code\*/

}

public static void main(String[] args) {

/\* Complete the code\*/

}

}

**Category**: Generic

Create a Printer class with a printArray() single method (No method overloading) that accept a single argument that can be String or integer array, and prints the array elements.

**Category**: Generic

Write a generic method to get the sum value of odd elements in a collection.

**Category**: Generic

Create a generic class in which we want to access the group of objects from same family.

Create 3 classes A,B and C which having the same method inside of all classes .A is a super class of B and C. Create generic class Bound and try to access the method inside the class using that Bound class.

**Category**: Collections

Write a method that accepts a Map object having key-value pairs of any size. The first value and the second value gets swapped. Similarly the next two should get interchange. For example

* If there are 5 keys and values like {"val1": "java", "val2": "c++",”val3”:”c#”,”val4”:”python”,”val5”:”php”}
* Then the output should be {"val1": "c++", "val2": "java",”val3”:”c#”,”val4”:”php”,”val5”:”python”}

***Conditions***

The size need not to be 5 it may be of any size.

It should swap the value of adjacent value.

If the size is odd then the value in the middle should not be swapped.

**Category**: Collections

From a given set of list, get the string values to be displayed in the order of the high number of vowels to the least number of vowels from the given string values. If the number of vowels are same then it should check for the length and display the minimum one first.

Example: if the values are length, adata, lost then the output should be as adata, lost, llength.

**Category**: Conditional Statements

Create a program that accepts a single character letter and prints whether it is a consonant or vowel.

Condition:

Print an error message for an if the input is not a letter

If the input having more than one letter, print the required output (Vowel or Consonant) for each letter

***If input is***:

‘p’  
***Expected Output*** is:  
Consonant

***If input is***:

‘ap’  
***Expected Output*** is:  
Vowel Consonant