Easy

1. Java If-Else
   https://www.hackerrank.com/challenges/java-if-else/problem

Task

Given an integer, n, perform the following conditional actions:

- If n is odd, print `Weird`
- If n is even and in the inclusive range of 2 to 5, print `Not Weird`
- If n is even and in the inclusive range of 6 to 20, print `Weird`
- If n is even and greater than 20, print `Not Weird`

Complete the stub code provided in your editor to print whether or not n is weird.

```
1.  import java.io.*;
2.  import java.math.*;
3.  import java.security.*;
4.  import java.text.*;
5.  import java.util.*;
6.  import java.util.concurrent.*;
7.  import java.util.regex.*;
8.
9.  public class Solution {
10.

11.     private static final Scanner scanner = new Scanner(System.in);
12.
13.     public static void main(String[] args) {
14.         int N = scanner.nextInt();
15.         scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");
16.
17.         scanner.close();
18.
19.         int mod = N % 2;
20.         if(mod == 1){
21.             System.out.println("Weird");
22.         }else if((mod == 0) && (N >= 2) && (N <= 5)){
23.             System.out.println("Not Weird");
24.         }else if((mod == 0) && (N > 6) && (N <= 20)){
25.             System.out.println("Weird");
26.         }else{
27.             System.out.println("Not Weird");
28.         }
29.     }
30. }
```

2. Java Stdin and Stdout II
   https://www.hackerrank.com/challenges/java-stdin-stdout/problem

Sample Input

```
42
3.1415
Welcome to HackerRank's Java tutorials!
```

Sample Output

```
String: Welcome to HackerRank's Java tutorials!
Double: 3.1415
Int: 42
```

```java
import java.util.Scanner;

public class Solution {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int i = scan.nextInt();
        double d = scan.nextDouble();
        scan.nextLine();
        String s = scan.nextLine();

        // Write your code here.

        System.out.println("String: " + s);
        System.out.println("Double: " + d);
        System.out.println("Int: " + i);
    }
}
```

3. Welcome to Java!
   https://www.hackerrank.com/challenges/welcome-to-java/problem

Input Format

There is no input for this challenge.

Output Format

You must print two lines of output:

1. Print `Hello, World.` on the first line.
2. Print `Hello, Java.` on the second line.

Sample Output

```
Hello, World.
Hello, Java.
```

```java
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Print output to STDOUT. Your class should be named Solution. */
        System.out.println("Hello, World.");
        System.out.println("Hello, Java.");
    }
}
```

4. Java End-of-file
   https://www.hackerrank.com/challenges/java-end-of-file/problem

Sample Input

```
Hello world
I am a file
Read me until end-of-file.
```

Sample Output

```
1 Hello world
2 I am a file
3 Read me until end-of-file.
```

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

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
        Scanner scan = new Scanner(System.in);
        int num = 1;

        while(scan.hasNextLine()){
            String s = scan.nextLine();

            System.out.println(num + " " + s);
            num = num + 1;
        }
    }
}
```

5. Pattern Syntax Checker
   https://www.hackerrank.com/challenges/pattern-syntax-checker/problem

Sample Input

```
3
([A-Z])(.+)
[AZ[a-z](a-z)
batcatpat(nat
```

Sample Output

```
Valid
Invalid
Invalid
```

```java
import java.util.Scanner;
import java.util.regex.*;

public class Solution
{
    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        int testCases = Integer.parseInt(in.nextLine());
        while(testCases>0){
            String pattern = in.nextLine();
            //Write your code
            try{
                Pattern pat = Pattern.compile(pattern);
                System.out.println("Valid");
            }catch(Exception e){
                System.out.println("Invalid");
            }

            testCases--;
        }
    }
}
```

6. Java Exception Handling (Try-catch)
   https://www.hackerrank.com/challenges/java-exception-handling-try-catch

Sample Input 0:

```
10
3
```

Sample Output 0:

```
3
```

Sample Input 1:

```
10
Hello
```

Sample Output 1:

```
java.util.InputMismatchException
```

Sample Input 2:

```
10
0
```

Sample Output 2:

```
java.lang.ArithmeticException: / by zero
```

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

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. */
        Scanner scan = new Scanner(System.in);

        try{
            int num1 = scan.nextInt();
            int num2 = scan.nextInt();
            int div = num1 / num2;
            System.out.println(div);
        }catch(InputMismatchException e){
            System.out.println(e.getClass().getName());
        }catch(ArithmeticException e){
            System.out.println(e);
        }
    }
}
```

7. Java Method Overriding
   https://www.hackerrank.com/challenges/java-method-overriding/problem

Task

Complete the code in your editor by writing an overridden getNumberOfTeamMembers method that prints the same statement as the superclass' getNumberOfTeamMembers method, except that it replaces n with 11 (the number of players on a Soccer team).

Output Format

When executed, your completed code should print the following:

```
Generic Sports
Each team has n players in Generic Sports
Soccer Class
Each team has 11 players in Soccer Class
```

```java
import java.util.*;
class Sports{

    String getName(){
        return "Generic Sports";
    }

    void getNumberOfTeamMembers(){
        System.out.println( "Each team has n players in " + getName() );
    }
}

class Soccer extends Sports{
    @Override
    String getName(){
        return "Soccer Class";
    }
    // Write your overridden getNumberOfTeamMembers method here
    void getNumberOfTeamMembers(){
        System.out.println("Each team has 11 players in " + getName());
    }
}

public class Solution{

    public static void main(String []args){
        Sports c1 = new Sports();
        Soccer c2 = new Soccer();
        System.out.println(c1.getName());
        c1.getNumberOfTeamMembers();
        System.out.println(c2.getName());
        c2.getNumberOfTeamMembers();
    }
}
```

8. Java Method Overriding 2 (Super Keyword)
   https://www.hackerrank.com/challenges/java-method-overriding-2-super-keyword/problem

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.
```

```java
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());

        BiCycle B = new BiCycle();
        String temp = B.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();
    }
}
```

Medium

1. Java Regex
   https://www.hackerrank.com/challenges/java-regex/problem

Sample Input

```
000.12.12.034
121.234.12.12
23.45.12.56
00.12.123.123123.123
122.23
Hello.IP
```

Sample Output

```
true
true
true
false
false
false
```

```java
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import java.util.Scanner;

class Solution{

    public static void main(String[] args){
        Scanner in = new Scanner(System.in);
        while(in.hasNext()){
            String IP = in.next();
            System.out.println(IP.matches(new MyRegex().pattern));
        }

    }
}

//Write your code here
class MyRegex{
    String part = "((25[0-5])|(2[0-4][0-9])|([0-1]{0,1}[0-9]{1,2}))";
    String pattern = part + "." + part + "." + part + "." + part;
}
```