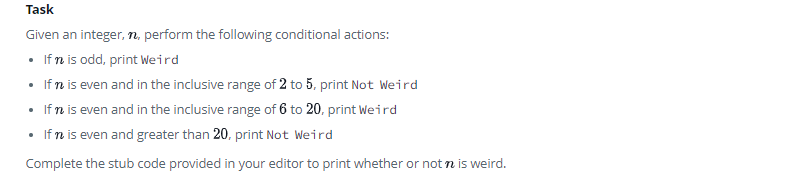
**Easy**

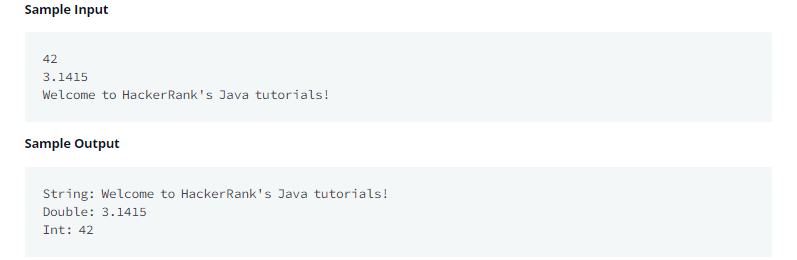
1. Java If-Else

<https://www.hackerrank.com/challenges/java-if-else/problem>



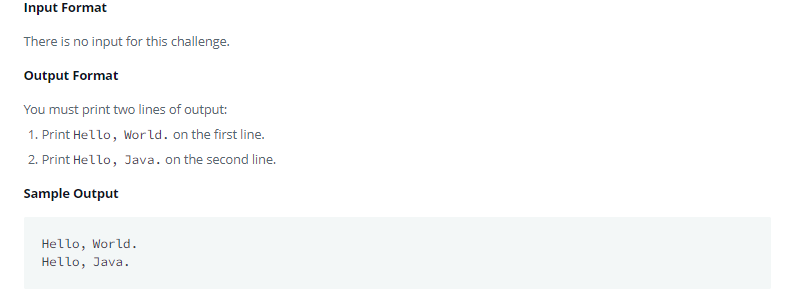
1. **import** java.io.\*;
2. **import** java.math.\*;
3. **import** java.security.\*;
4. **import** java.text.\*;
5. **import** java.util.\*;
6. **import** java.util.concurrent.\*;
7. **import** java.util.regex.\*;
8. **public** **class** Solution {
10. **private** **static** **final** Scanner scanner = **new** Scanner(System.in);
11. **public** **static** **void** main(String[] args) {
12. **int** N = scanner.nextInt();
13. scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");
14. scanner.close();
15. **int** mod = N % 2;
16. **if**(mod == 1){
17. System.out.println("Weird");
18. }**else** **if**((mod == 0) && (N >= 2) && (N <= 5)){
19. System.out.println("Not Weird");
20. }**else** **if**((mod == 0) && (N > 6) && (N <= 20)){
21. System.out.println("Weird");
22. }**else**{
23. System.out.println("Not Weird");
24. }
25. }
26. }
27. Java Stdin and Stdout II

<https://www.hackerrank.com/challenges/java-stdin-stdout/problem>



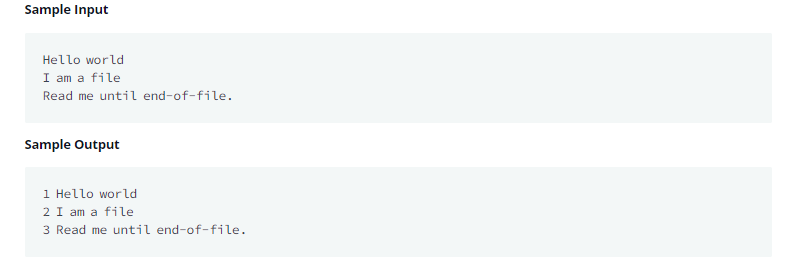
1. **import** java.util.Scanner;
2. **public** **class** Solution {
3. **public** **static** **void** main(String[] args) {
4. Scanner scan = **new** Scanner(System.in);
5. **int** i = scan.nextInt();
6. **double** d = scan.nextDouble();
7. scan.nextLine();
8. String s = scan.nextLine();
9. *// Write your code here.*
10. System.out.println("String: " + s);
11. System.out.println("Double: " + d);
12. System.out.println("Int: " + i);
13. }
14. }
15. Welcome to Java!

<https://www.hackerrank.com/challenges/welcome-to-java/problem>



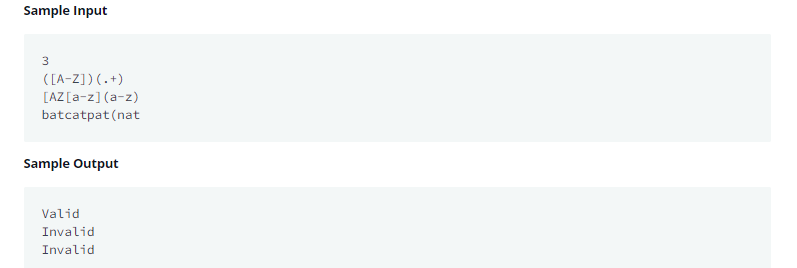
1. **public** **class** Solution {
2. **public** **static** **void** main(String[] args) {
3. */\* Enter your code here. Print output to STDOUT. Your class should be named Solution. \*/*
4. System.out.println("Hello, World.");
5. System.out.println("Hello, Java.");
6. }
7. }
8. Java End-of-file

<https://www.hackerrank.com/challenges/java-end-of-file/problem>



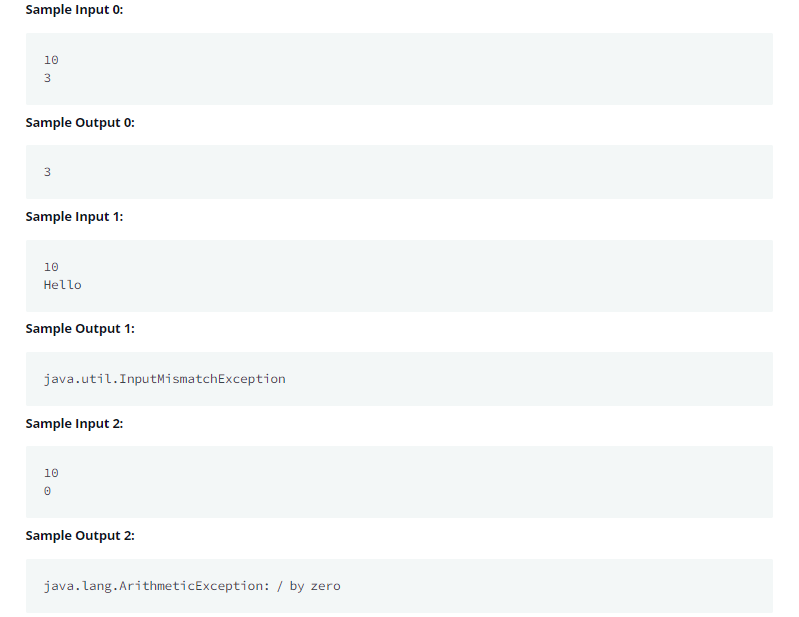
1. **import** java.io.\*;
2. **import** java.util.\*;
3. **public** **class** Solution {
4. **public** **static** **void** main(String[] args) {
5. */\* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. \*/*
6. Scanner scan = **new** Scanner(System.in);
7. **int** num = 1;
9. **while**(scan.hasNextLine()){
10. String s = scan.nextLine();
11. System.out.println(num + " " + s);
12. num = num + 1;
13. }
14. }
15. }
16. Pattern Syntax Checker

<https://www.hackerrank.com/challenges/pattern-syntax-checker/problem>



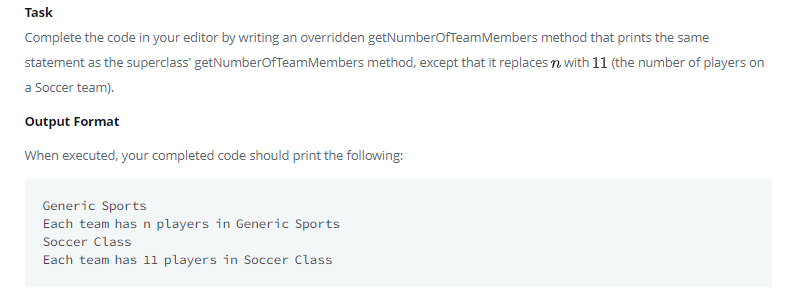
1. **import** java.util.Scanner;
2. **import** java.util.regex.\*;
3. **public** **class** Solution
4. {
5. **public** **static** **void** main(String[] args){
6. Scanner in = **new** Scanner(System.in);
7. **int** testCases = Integer.parseInt(in.nextLine());
8. **while**(testCases>0){
9. String pattern = in.nextLine();
10. *//Write your code*
11. **try**{
12. Pattern pat = Pattern.compile(pattern);
13. System.out.println("Valid");
14. }**catch**(Exception e){
15. System.out.println("Invalid");
16. }
17. testCases--;
18. }
19. }
20. }
21. Java Exception Handling (Try-catch)

<https://www.hackerrank.com/challenges/java-exception-handling-try-catch>



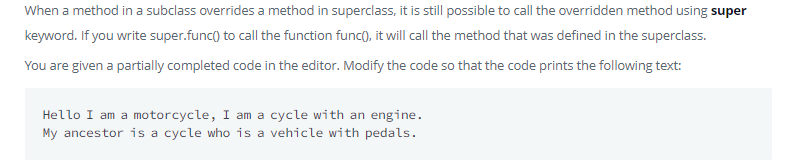
1. **import** java.io.\*;
2. **import** java.util.\*;
3. **public** **class** Solution {
4. **public** **static** **void** main(String[] args) {
5. */\* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. \*/*
6. Scanner scan = **new** Scanner(System.in);
7. **try**{
8. **int** num1 = scan.nextInt();
9. **int** num2 = scan.nextInt();
10. **int** div = num1 / num2;
11. System.out.println(div);
12. }**catch**(InputMismatchException e){
13. System.out.println(e.getClass().getName());
14. }**catch**(ArithmeticException e){
15. System.out.println(e);
16. }
17. }
18. }
19. Java Method Overriding

<https://www.hackerrank.com/challenges/java-method-overriding/problem>



1. **import** java.util.\*;
2. **class** Sports{
3. String getName(){
4. **return** "Generic Sports";
5. }
7. **void** getNumberOfTeamMembers(){
8. System.out.println( "Each team has n players in " + getName() );
9. }
10. }
11. **class** Soccer **extends** Sports{
12. @Override
13. String getName(){
14. **return** "Soccer Class";
15. }
16. *// Write your overridden getNumberOfTeamMembers method here*
17. **void** getNumberOfTeamMembers(){
18. System.out.println("Each team has 11 players in " + getName());
19. }
20. }
21. **public** **class** Solution{
23. **public** **static** **void** main(String []args){
24. Sports c1 = **new** Sports();
25. Soccer c2 = **new** Soccer();
26. System.out.println(c1.getName());
27. c1.getNumberOfTeamMembers();
28. System.out.println(c2.getName());
29. c2.getNumberOfTeamMembers();
30. }
31. }
32. Java Method Overriding 2 (Super Keyword)

<https://www.hackerrank.com/challenges/java-method-overriding-2-super-keyword/problem>

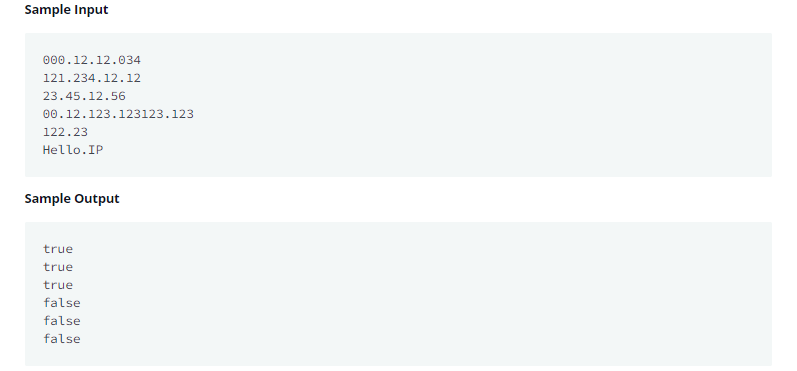


1. **import** java.util.\*;
2. **import** java.io.\*;
3. **class** BiCycle{
4. String define\_me(){
5. **return** "a vehicle with pedals.";
6. }
7. }
8. **class** MotorCycle **extends** BiCycle{
9. String define\_me(){
10. **return** "a cycle with an engine.";
11. }
13. MotorCycle(){
14. System.out.println("Hello I am a motorcycle, I am "+ define\_me());
15. BiCycle B = **new** BiCycle();
16. String temp = B.define\_me(); *//Fix this line*
17. System.out.println("My ancestor is a cycle who is "+ temp );
18. }
20. }
21. **class** Solution{
22. **public** **static** **void** main(String []args){
23. MotorCycle M=**new** MotorCycle();
24. }
25. }

**Medium**

1. Java Regex

<https://www.hackerrank.com/challenges/java-regex/problem>



1. **import** java.util.regex.Matcher;
2. **import** java.util.regex.Pattern;
3. **import** java.util.Scanner;
4. **class** Solution{
5. **public** **static** **void** main(String[] args){
6. Scanner in = **new** Scanner(System.in);
7. **while**(in.hasNext()){
8. String IP = in.next();
9. System.out.println(IP.matches(**new** MyRegex().pattern));
10. }
11. }
12. }
13. *//Write your code here*
14. **class** MyRegex{
15. String part = "((25[0-5])|(2[0-4][0-9])|([0-1]{0,1}[0-9]{1,2}))";
16. String pattern = part + "." + part + "." + part + "." + part;
17. }