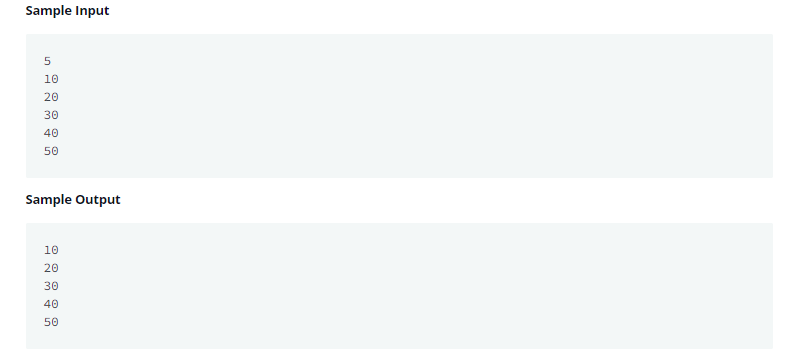
**Easy**

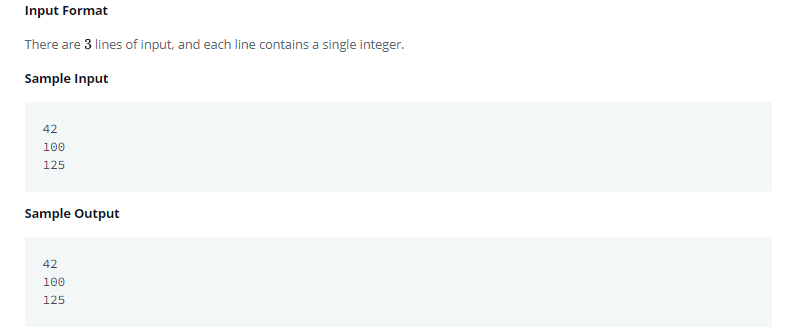
1. Java 1D Array

<https://www.hackerrank.com/challenges/java-1d-array-introduction/problem>



1. **import** java.util.\*;
2. **public** **class** Solution {
3. **public** **static** **void** main(String[] args) {
5. Scanner scan = **new** Scanner(System.in);
6. **int** n = scan.nextInt();
7. **int**[] a = **new** **int**[n];
8. **for**(**int** i = 0; i < n; i++){
9. **int** num = scan.nextInt();
10. a[i] = num;
11. }
12. scan.close();
13. *// Prints each sequential element in array a*
14. **for** (**int** i = 0; i < a.length; i++) {
15. System.out.println(a[i]);
16. }
17. }
18. }
19. Java Stdin and Stdout I

<https://www.hackerrank.com/challenges/java-stdin-and-stdout-1/problem>

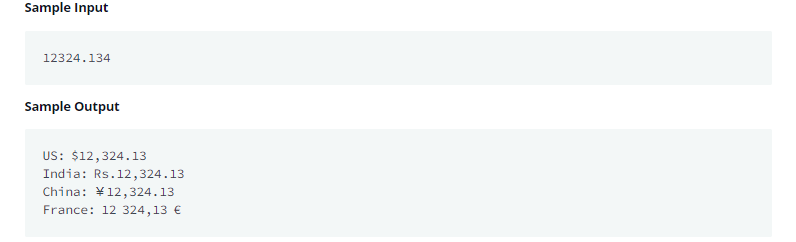


1. **import** java.util.\*;
2. **public** **class** Solution {
3. **public** **static** **void** main(String[] args) {
4. Scanner scan = **new** Scanner(System.in);
5. **int** a = scan.nextInt();
6. **int** b = scan.nextInt();
7. **int** c = scan.nextInt();
8. *// Complete this line*
9. *// Complete this line*
10. System.out.println(a);
11. System.out.println(b);
12. System.out.println(c);
13. *// Complete this line*
14. *// Complete this line*
15. }
16. }
17. Java Inheritance I

<https://www.hackerrank.com/challenges/java-inheritance-1/problem>

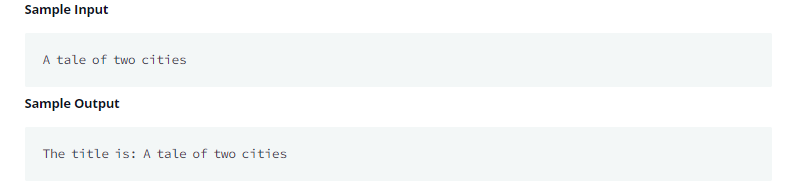
1. **import** java.io.\*;
2. **import** java.util.\*;
3. **import** java.text.\*;
4. **import** java.math.\*;
5. **import** java.util.regex.\*;
6. **class** Animal{
7. **void** walk()
8. {
9. System.out.println("I am walking");
10. }
11. }
12. **class** Bird **extends** Animal
13. {
14. **void** fly()
15. {
16. System.out.println("I am flying");
17. }
18. **void** sing(){
19. System.out.println("I am singing");
20. }
21. }
22. **public** **class** Solution{
23. **public** **static** **void** main(String args[]){
24. Bird bird = **new** Bird();
25. bird.walk();
26. bird.fly();
27. bird.sing();
29. }
30. }
31. Java Currency Formatter

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



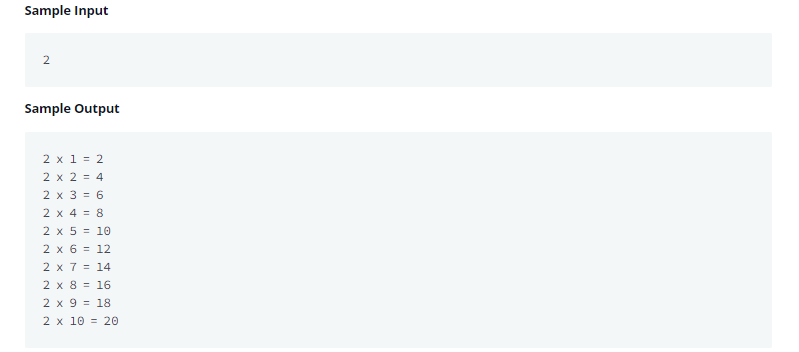
1. **import** java.io.\*;
2. **import** java.util.\*;
3. **import** java.text.\*;
4. **import** java.math.\*;
5. **import** java.util.regex.\*;
6. **public** **class** Solution {
8. **public** **static** **void** main(String[] args) {
9. Scanner scanner = **new** Scanner(System.in);
10. **double** payment = scanner.nextDouble();
11. scanner.close();
12. *// Write your code here.*
13. Locale indiaLocale = **new** Locale("en", "IN");
14. NumberFormat us = NumberFormat.getCurrencyInstance(Locale.US);
15. NumberFormat india = NumberFormat.getCurrencyInstance(indiaLocale);
16. NumberFormat china = NumberFormat.getCurrencyInstance(Locale.CHINA);
17. NumberFormat france = NumberFormat.getCurrencyInstance(Locale.FRANCE);
19. System.out.println("US: " + us.format(payment));
20. System.out.println("India: " + india.format(payment));
21. System.out.println("China: " + china.format(payment));
22. System.out.println("France: " + france.format(payment));
23. }
24. }
25. Java Abstract Class

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



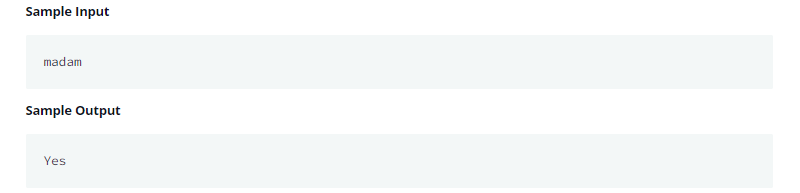
1. **import** java.util.\*;
2. **abstract** **class** Book{
3. String title;
4. **abstract** **void** setTitle(String s);
5. String getTitle(){
6. **return** title;
7. }
9. }
10. *//Write MyBook class here*
11. **class** MyBook **extends** Book{
12. **void** setTitle(String s){
13. title = s;
14. }
15. }
16. **public** **class** Main{
18. **public** **static** **void** main(String []args){
19. *//Book new\_novel=new Book(); This line prHMain.java:25: error: Book is abstract; cannot be instantiated*
20. Scanner sc=**new** Scanner(System.in);
21. String title=sc.nextLine();
22. MyBook new\_novel=**new** MyBook();
23. new\_novel.setTitle(title);
24. System.out.println("The title is: "+new\_novel.getTitle());
25. sc.close();
27. }
28. }
29. Java Loops I

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



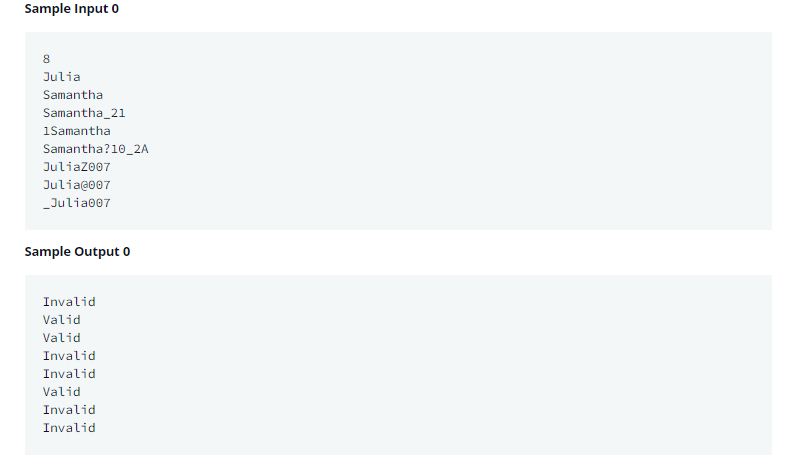
1. **import** java.io.\*;
2. **import** java.math.\*;
3. **import** java.security.\*;
4. **import** java.text.\*;
5. **import** java.util.\*;
6. **import** java.util.concurrent.\*;
7. **import** java.util.regex.\*;
8. **public** **class** Solution {
10. **private** **static** **final** Scanner scanner = **new** Scanner(System.in);
11. **public** **static** **void** main(String[] args) {
12. **int** N = scanner.nextInt();
13. **for**(**int** x = 1; x <= 10; x++){
14. **int** Hasil = N \* x;
15. System.out.println(N +" x "+ x +" = "+ Hasil);
16. }
17. scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");
18. scanner.close();
19. }
20. }
21. Java String Reverse

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



1. **import** java.io.\*;
2. **import** java.util.\*;
3. **public** **class** Solution {
4. **public** **static** **void** main(String[] args) {
6. Scanner sc=**new** Scanner(System.in);
7. String A=sc.next();
8. */\* Enter your code here. Print output to STDOUT. \*/*
9. String B = "";
10. **for**(**int** i = A.length() - 1; i >= 0; i--){
11. B = B + A.charAt(i);
12. }
13. **if**(A.equals(B)){
14. System.out.print("Yes");
15. }**else**{
16. System.out.print("No");
17. }
19. }
20. }
21. Valid Username Regular Expression

<https://www.hackerrank.com/challenges/valid-username-checker/problem>

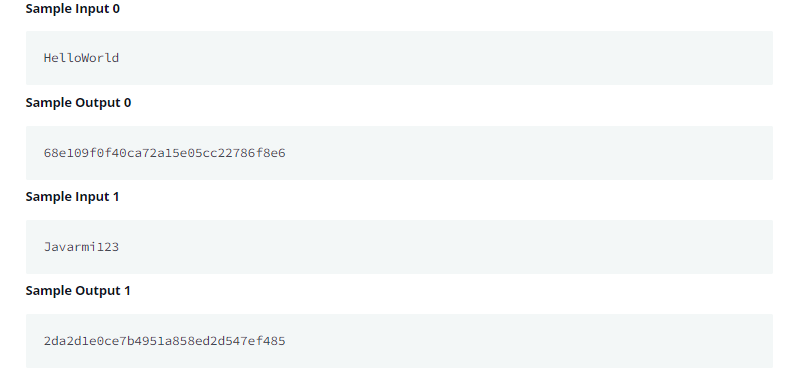


1. **import** java.util.Scanner;
2. **class** UsernameValidator {
3. */\**
4. *\* Write regular expression here.*
5. *\*/*
6. **public** **static** **final** String regularExpression = "^[a-zA-Z][\\w]{7,29}$";
7. }
8. **public** **class** Solution {
9. **private** **static** **final** Scanner scan = **new** Scanner(System.in);
11. **public** **static** **void** main(String[] args) {
12. **int** n = Integer.parseInt(scan.nextLine());
13. **while** (n-- != 0) {
14. String userName = scan.nextLine();
15. **if** (userName.matches(UsernameValidator.regularExpression)) {
16. System.out.println("Valid");
17. } **else** {
18. System.out.println("Invalid");
19. }
20. }
21. }
22. }

**Medium**

1. Java MD5

<https://www.hackerrank.com/challenges/java-md5/problem>



1. **import** java.io.\*;
2. **import** java.util.\*;
3. **import** java.security.\*;
4. **public** **class** Solution {
5. **public** **static** **void** main(String[] args) **throws** NoSuchAlgorithmException {
6. */\* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named Solution. \*/*
7. Scanner scan = **new** Scanner(System.in);
8. String word = scan.nextLine();
9. MessageDigest md = MessageDigest.getInstance("MD5");
10. **byte**[] hashInBytes = md.digest(word.getBytes());
11. StringBuilder sb = **new** StringBuilder();
12. **for**(**byte** b : hashInBytes){
13. sb.append(String.format("%02x", b));
14. }
15. System.out.println(sb.toString());
16. }
17. }