// 1. Suppose, a=10 and b=20. Now swap the value using a temp variable. Output: a=20, b=10

public class SwapValueUsingTempVariable {

public static void main(String[] args) {

int a = 10;

int b = 20;

System.*out*.println("Given Output:");

System.*out*.println("a=" + a);

System.*out*.println("b=" + b);

int temp = a;

a = b;

b = temp;

System.*out*.println("After Done Swap:");

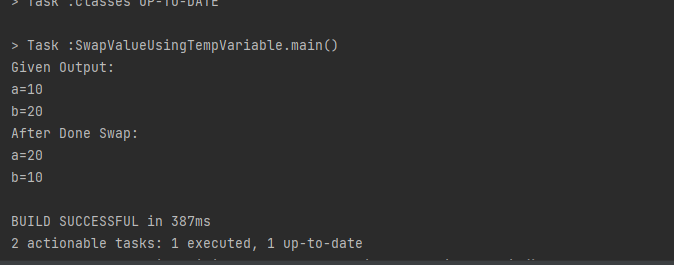
System.*out*.println("a=" + a);

System.*out*.println("b=" + b);

}

}

Output:



// 2. Suppose, a=10 and b=20. Now swap the value without a temp variable. Output: a=20, b=10

public class SwapValueWithoutUsingTempVariable {

public static void main(String[] args) {

int a = 10;

int b = 20;

System.*out*.println("Given Output:");

System.*out*.println("a=" + a);

System.*out*.println("b=" + b);

int b1 = a + b;

a = b1 - a;

b = b1 - b;

System.*out*.println("After Done Swap:");

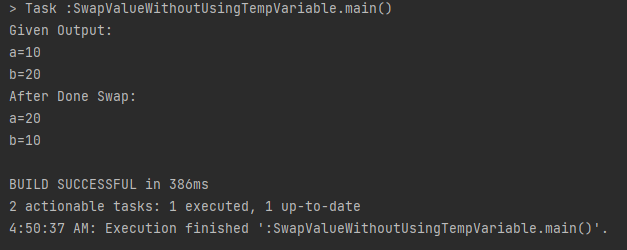
System.*out*.println("a=" + a);

System.*out*.println("b=" + b);

}

}

Output:



//3. Suppose a=15.5276. print the value upto 2 decimal point. Output: a=15.52

public class Print2DecimalValue {

public static void main(String[] args) {

double a = 15.5276;

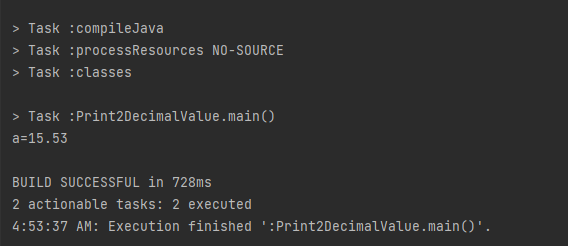
double twoValue = Math.*round*(a\*100.0)/100.0;

System.*out*.printf("a=%.2f%n", twoValue);

}

}

Output:



// 4. Write a program that will find your key is found in the given array using binary search method numbers=[1,6,9,3,5,4,7]

//key=5

import java.util.Arrays;

public class BinarySearch {

public static void main(String args[]){

int arr[] = {1, 6, 9, 3, 5, 4, 7};

int key = 5;

Arrays.*sort*(arr);

int last=arr.length-1;

*binarySearch*(arr,0,last,key);

}

public static void binarySearch(int arr[], int first, int last, int key){

int mid = (first + last)/2;

while( first <= last ){

if ( arr[mid] < key ){

first = mid + 1;

}else if ( arr[mid] == key ){

System.*out*.println("Key found at index: " + mid);

break;

}else{

last = mid - 1;

}

mid = (first + last)/2;

}

if ( first > last ){

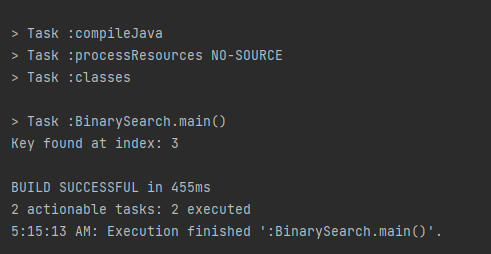
System.*out*.println("Key not found!");

}

}

}

Output:



// 5. Write a program that will find your key is found in the given array using linear search method numbers=[1,6,9,3,5,4,7]

//key=5

public class LinearSearch {

public static void main(String[] args) {

int[] n = {1, 6, 9, 3, 5, 4, 7};

int k = 5;

int index = *linearSearch*(n, k);

if (index != -1) {

System.*out*.println("Key found at index: " + index);

} else {

System.*out*.println("Key not found");

}

}

public static int linearSearch(int[] arr, int k) {

for (int i = 0; i < arr.length; i++) {

if (arr[i] == k) {

return i;

}

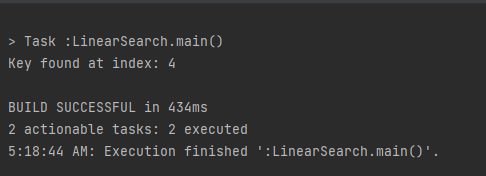
}

return -1;

}

}

Output:



// 6. Generate random 10 integer numbers in an array and print out all the numbers from array and also print the max and min number from the array

import java.util.Arrays;

import java.util.Random;

public class TenRandomNumbersMaxMin {

public static void main(String[] args) {

int[] numbers = *generateRandomArray*(10);

System.*out*.println("10 Random Numbers: " + Arrays.*toString*(numbers));

int max = *findMaxNumber*(numbers);

System.*out*.println("The Maximum: " + max);

int minNumber = *findMinNumber*(numbers);

System.*out*.println("The Minimum: " + minNumber);

}

public static int[] generateRandomArray(int size) {

int[] array = new int[size];

Random random = new Random();

for (int i = 0; i < size; i++) {

array[i] = random.nextInt();

}

return array;

}

public static int findMaxNumber(int[] array) {

int max = array[0];

for (int i = 1; i < array.length; i++) {

if (array[i] > max) {

max = array[i];

}

}

return max;

}

public static int findMinNumber(int[] array) {

int min = array[0];

for (int i = 1; i < array.length; i++) {

if (array[i] < min) {

min = array[i];

}

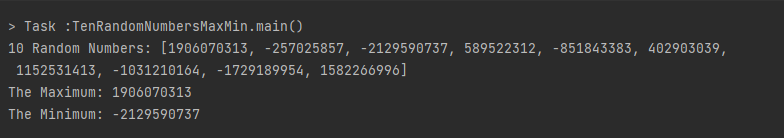
}

return min;

}

}

Output:



// 7. Write a program that will breakdown the amount and count notes for any given amount. Here is the notes in the given array:

//notes=[1000,500,200,100,50,20,10,5,2,1]

//

//Example 1: User input: 575

//Output:

//500 1

//50 1

//20 1

//5 1

//Example 2: User input: 2148

//Output:

//1000 2

//100 1

//20 2

//5 1

//2 1

//1 1

import java.util.Scanner;

public class BreakdownTheAmountIntoNotes {

public static void main(String[] args) {

int[] notes = {1000, 500, 200, 100, 50, 20, 10, 5, 2, 1};

Scanner scanner = new Scanner(System.*in*);

System.*out*.print("Enter any amount for breakdown: ");

int amount = scanner.nextInt();

scanner.close();

System.*out*.println("Breakdown Notes:");

for (int note : notes) {

if (amount >= note) {

int count = amount / note;

System.*out*.println(note + " " + count);

amount %= note;

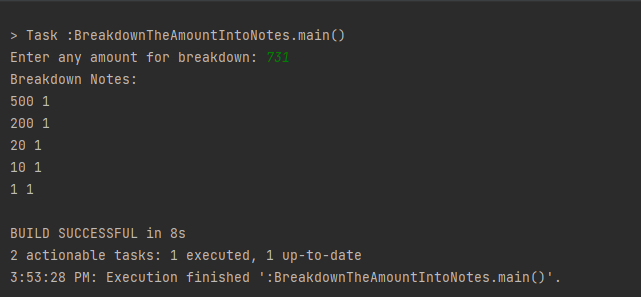
}

}

}

}

Output:



// 8. Print the prime numbers of 2 to 100

public class PrimeNumber2to100 {

public static void main(String[] args) {

System.*out*.println("Found prime number 2 to 100 is:");

for (int N = 2; N <= 100; N++) {

boolean isPrime = true;

for (int i = 2; i <= Math.*sqrt*(N); i++) {

if (N % i == 0) {

isPrime = false;

break;

}

}

if (isPrime) {

System.*out*.println(N);

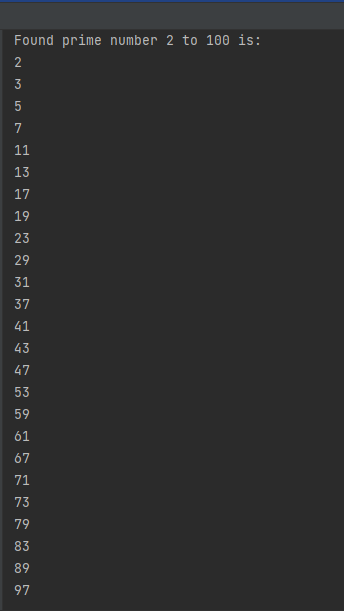
}

}

}

}

Output:



// 9. Write a program that will give following output:

//1

//12

//123

//1234

//12345

public class NumberPattern1 {

public static void main(String[] args) {

int rows = 5;

for (int i = 1; i <= rows; i++) {

for (int j = 1; j <= i; j++) {

System.*out*.print(j);

}

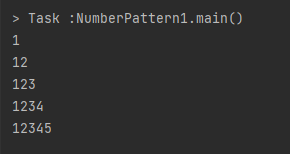
System.*out*.println();

}

}

}

Output:



// 10. Write a program that will give following output:

//12345

//2345

//345

//45

//5

public class NumberPattern2 {

public static void main(String[] args) {

int n = 12345;

String N = Integer.*toString*(n);

for (int i = 0; i < N.length(); i++)

{

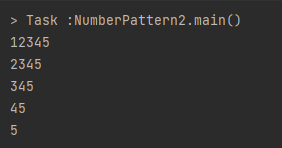
System.*out*.println(N.substring(i));

}

}

}

Output:



// 11. Find out the second largest element of the given array numbers=[5,3,9,7,4,1,8]

public class FindTheSecondLargest {

public static void main(String[] args) {

int[] numbers = {5, 3, 9, 7, 4, 1, 8};

int secondLargest = *findSecondLargest*(numbers);

System.*out*.println("Second Largest Number is: " + secondLargest);

}

public static int findSecondLargest(int[] arr) {

int largest = Integer.*MIN\_VALUE*;

int secondLargest = Integer.*MIN\_VALUE*;

for (int i = 0; i < arr.length; i++) {

if (arr[i] > largest) {

secondLargest = largest;

largest = arr[i];

} else if (arr[i] > secondLargest && arr[i] < largest) {

secondLargest = arr[i];

}

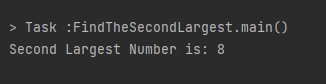
}

return secondLargest;

}

}

Output:



// 12. Write a program that will shuffle (values will randomly change their position) from the given array numbers=[1,2,3,4,5,6,7,8,9,0]

//random output: 3 5 0 1 7 9 6 4 2 8

import java.util.Arrays;

import java.util.Random;

public class ShuffleTheNumbersPosition {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};

*shuffleArray*(numbers);

System.*out*.println("After the shuffle: " + Arrays.*toString*(numbers));

}

public static void shuffleArray(int[] array) {

Random random = new Random();

for (int i = array.length - 1; i > 0; i--) {

int j = random.nextInt(i + 1);

int temp = array[i];

array[i] = array[j];

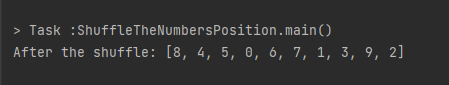
array[j] = temp;

}

}

}

Output:



// 13. Take any number as input and print the reverse of the number input: 12345

//output: 54321

import java.util.Scanner;

public class ReverseTheInputNumber {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.*in*);

System.*out*.print("Enter any Number: ");

int number = scanner.nextInt();

int R = 0;

while (number != 0) {

int digit = number % 10;

R = R \* 10 + digit;

number /= 10;

}

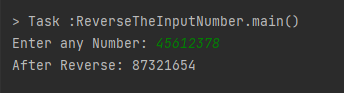
System.*out*.println("After Reverse: " + R);

scanner.close();

}

}

Output:



// 14. Find out the duplicate numbers in the given array and in which position they are found. numbers=[1,2,3,1,2,4,5,6,4]

//Output:

//Duplicate value: 1 at index 3

//Duplicate value: 2 at index 4

//Duplicate value: 4 at index 8

import java.util.HashMap;

public class FindTheDuplicateValue {

public static void main(String[] args) {

int[] numbers = {1, 2, 3, 1, 2, 4, 5, 6, 4};

HashMap<Integer, Integer> numberMap = new HashMap<>();

for (int i = 0; i < numbers.length; i++) {

int number = numbers[i];

if (numberMap.containsKey(number)) {

System.*out*.println("Duplicate value: " + number + " at index " + i);

} else {

numberMap.put(number, i);

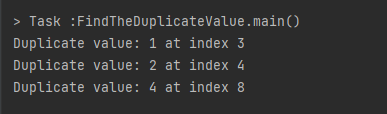
}

}

}

}

Output:



// 15. Count number of words, number of characters without spaces, number of vowels and consonant from the given string:

//"I live in Bangladesh"

//Output:

//Number of words: 4

//Number of chars without spaces: 17

//Number of vowels: 7

//Number of consonant: 10

public class CountTheNumberOfWords {

public static void main(String[] args) {

String input = "I live in Bangladesh";

String[] words = input.split("\\s+");

int wordCount = words.length;

int charCountWithoutSpaces = input.replace(" ", "").length();

int vowelCount = 0;

int consonantCount = 0;

String lowercaseInput = input.toLowerCase();

for (int i = 0; i < lowercaseInput.length(); i++) {

char ch = lowercaseInput.charAt(i);

if (Character.*isLetter*(ch)) {

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

vowelCount++;

} else {

consonantCount++;

}

}

}

System.*out*.println("Number of words: " + wordCount);

System.*out*.println("Number of characters without spaces: " + charCountWithoutSpaces);

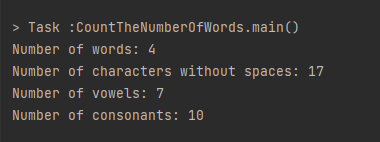
System.*out*.println("Number of vowels: " + vowelCount);

System.*out*.println("Number of consonants: " + consonantCount);

}

}

Output:



// 16. Encrypt word: ROADTOSDET [when A=F]

//Output: WTFIYTXIJY

public class EncryptWord {

public static void main(String[] args) {

String word = "ROADTOSDET";

String encryptedWord = *encrypt*(word, 'A', 'F');

System.*out*.println("Encrypted word: " + encryptedWord);

}

public static String encrypt(String word, char originalChar, char replacementChar) {

StringBuilder encryptedWord = new StringBuilder();

for (int i = 0; i < word.length(); i++) {

char currentChar = word.charAt(i);

if (Character.*isLetter*(currentChar)) {

char encryptedChar = (char) (currentChar + (replacementChar - originalChar));

if (encryptedChar > 'Z') {

encryptedChar = (char) ('A' + (encryptedChar - 'Z' - 1));

}

encryptedWord.append(encryptedChar);

} else {

encryptedWord.append(currentChar);

}

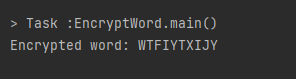
}

return encryptedWord.toString();

}

}

Output:



// 17. Check if the given string is palindrome or not. Palindrome means after reversing a string, it will be same.

//Input: Civic

//Output: true

//Input: One

//Output: false

public class PalindromeCheck {

public static boolean isPalindrome(String input) {

String reversed = new StringBuilder(input).reverse().toString();

return input.equalsIgnoreCase(reversed);

}

public static void main(String[] args) {

String input1 = "Civic";

System.*out*.println("Input: " + input1);

System.*out*.println("Output: " + *isPalindrome*(input1));

String input2 = "One";

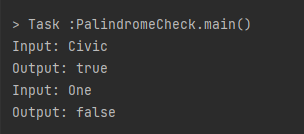
System.*out*.println("Input: " + input2);

System.*out*.println("Output: " + *isPalindrome*(input2));

}

}

Output:



// 18. Write a program that will take integer numbers as user input continuously and print the sum of numbers until user input q from the keyboard. When user input q, program will be quit. If user inputs another character, then the program will ask to input the number again.

import java.util.Scanner;

public class ContinueSumUserInput {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.*in*);

int sum = 0;

while (true) {

System.*out*.print("Enter numbers for sum (Input q when quit): ");

String input = scanner.nextLine();

if (input.equalsIgnoreCase("q")) {

break;

}

try {

int number = Integer.*parseInt*(input);

sum += number;

} catch (NumberFormatException e) {

System.*out*.println("Not a valid number. Please enter any valid number or input 'q' for quit ");

}

}

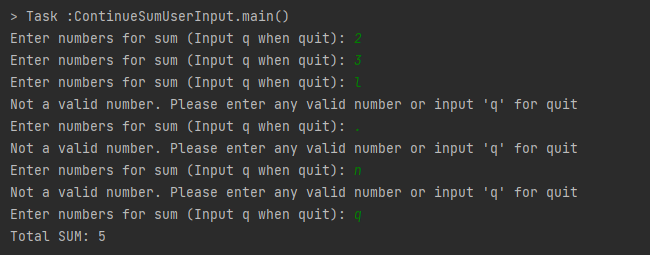
System.*out*.println("Total SUM: " + sum);

scanner.close();

}

}

Output:



// 19. Extract the OTP from the SMS. "Your one time password is 246148. Don't share this code with anyone \r\nBvwt3f8js2S"

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class OTPExtractor {

private static final Pattern *OTP\_PATTERN* = Pattern.*compile*("Your one time password is (.{6})");

public static String extractOTP(String sms) {

Matcher matcher = *OTP\_PATTERN*.matcher(sms);

if (matcher.find()) {

return matcher.group(1);

} else {

return null;

}

}

public static void main(String[] args) {

String sms = "Your one time password is 246148. Don't share this code with anyone \r\nBvwt3f8js2S";

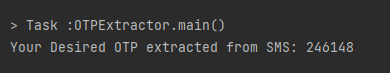
String otp = *extractOTP*(sms);

System.*out*.println("Your Desired OTP extracted from SMS: " + otp);

}

}

Output:



// 20. A core i7 laptop price is 85000 tk and a gaming mouse price is 2500 tk. If I buy the laptop and 1 piece mouse, what will be my total cost after giving 15% discount? [Extract the digits from the paragraph and calculate the price]

public class CalculatePriceAfterDiscount {

public static void main(String[] args) {

String paragraph = "A core i7 laptop price is 85000 tk and a gaming mouse price is 2500 tk.";

String[] words = paragraph.split("\\s+");

int laptopPrice = 0;

int mousePrice = 0;

for (String word : words) {

if (word.matches("\\d+")) {

int price = Integer.*parseInt*(word);

if (laptopPrice == 0) {

laptopPrice = price;

} else {

mousePrice = price;

break;

}

}

}

double discountPercentage = 15.0; // 15%

double discount = discountPercentage / 100.0;

double laptopCost = laptopPrice - (laptopPrice \* discount);

double mouseCost = mousePrice - (mousePrice \* discount);

double totalCost = laptopCost + mouseCost;

System.*out*.println("Laptop Price after 15% discount: " + laptopCost + " tk");

System.*out*.println("Mouse Price after 15% discount: " + mouseCost + " tk");

System.*out*.println("Total (Laptop+Mouse) cost after 15% discount: " + totalCost + " tk");

}

}

Output:

