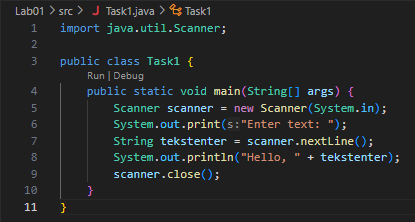
**ЛАБОРАТОРНА РОБОТА №1**

**Типи даних, умови, цикли та масиви**

**Мета роботи:** Використовуючи теоретичне підґрунтя про типи даних, що використовуються у мові Java, правила використання умовних переходів, написання циклів та використання масивів виконати дії що будуть вказано в завданні до лабораторної роботи.

**Завдання №1.** Напишіть програму, яка прочитає рядок із System.in і виведе повідомлення: "Hello, прочитаний рядок". Якщо у вхідному рядку кілька слів, потрібно надрукувати їх усі.

**Хід виконання завдання:**

****

**Вивід:**

****

****

**Текст програми:**

|  |
| --- |
| import java.util.Scanner;  public class Task1 {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.print("Enter text: ");  String tekstenter = scanner.nextLine();  System.out.println("Hello, " + tekstenter);  scanner.close();  }  } |

**Завдання №2.** Напишіть програму, яка отримує на вхід ціле число. Це число є поточний добовий час, який обчислюється як кількість секунд, що пройшли з півночі. Потім програма повинна надрукувати значення екрану електронного годинника для цього часу.

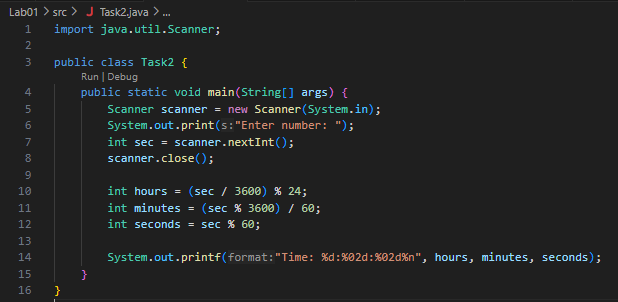
Вхідне значення отримується через System.in. Вихідне значення має бути надруковане у System.out. Гарантується, що вхідне значення — невід'ємне. Формат виводу: h:mm:ss (можливі значення: [0:00:00; 23:59:59]).

Приклад:

Введення даних: 60

Результат виконання: 0:01:00

**Хід виконання завдання:**

****

**Вивід:**

****

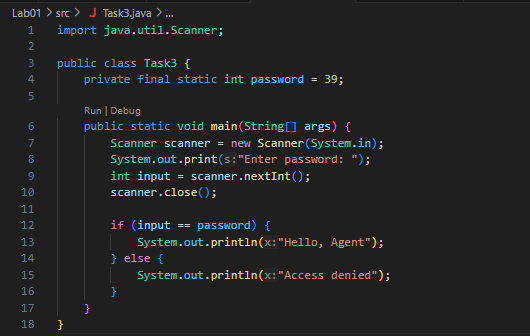
**Текст програми:**

|  |
| --- |
| import java.util.Scanner;  public class Task2 {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.print("Enter number: ");  int sec = scanner.nextInt();  scanner.close();  int hours = (sec / 3600) % 24;  int minutes = (sec % 3600) / 60;  int seconds = sec % 60;  System.out.printf("Time: %d:%02d:%02d%n", hours, minutes, seconds);  }  } |

**Завдання №3.** Напишіть програму, яка запитує на вхід число-пароль. Якщо воно дорівнює секретній кількості, програма виводить: "Hello, Agent", інакше друкує: "Access denied".

Секретний пароль зберігається в final static int password.

**Хід виконання завдання:**

****

**Вивід:**

****

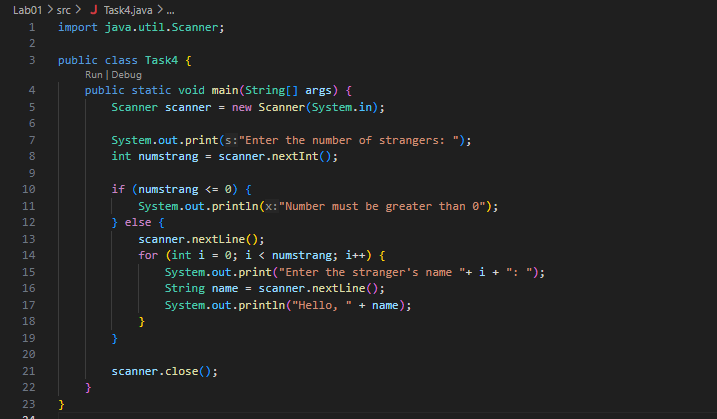
****

**Текст програми:**

|  |
| --- |
| import java.util.Scanner;  public class Task3 {  private final static int password = 39;    public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.print("Enter password: ");  int input = scanner.nextInt();  scanner.close();  if (input == password) {  System.out.println("Hello, Agent");  } else {  System.out.println("Access denied");  }  }  } |

**Завдання №4.** Напишіть просту програму, яка запитує кількість незнайомців, з якими потрібно зустрітися, рядково читає імена незнайомців, по рядках виводить: "Hello, (ім'я незнайомця)" для кожного незнайомця. Доповнити програму перевірками на кількість, що є від’ємною, та рівна 0.

**Хід виконання завдання:**

****

**Вивід:**

****

****

**Текст програми:**

|  |
| --- |
| import java.util.Scanner;  public class Task4 {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.print("Enter the number of strangers: ");  int numstrang = scanner.nextInt();  if (numstrang <= 0) {  System.out.println("Number must be greater than 0");  } else {  scanner.nextLine();  for (int i = 0; i < numstrang; i++) {  System.out.print("Enter the stranger's name "+ i + ": ");  String name = scanner.nextLine();  System.out.println("Hello, " + name);  }  }  scanner.close();  }  } |

**Завдання №5.** Наведено таку умову: равлик піднімається по дереву на a футів на день. Потім щоночі равлик сповзає вниз на b футів. Висота дерева — h футів.

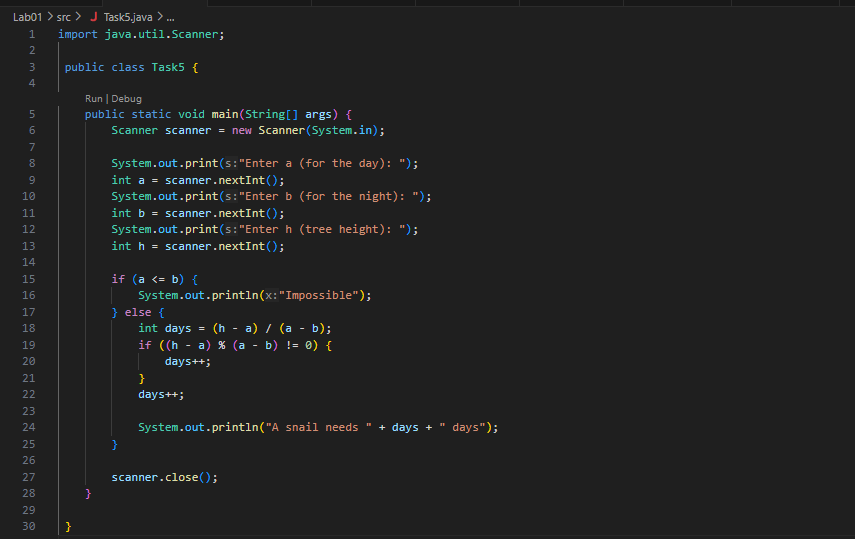
Напишіть програму з підрахунком кількості днів, які знадобляться равлику, щоб дістатися до вершини дерева. Програма читає a, b, h рядково. Вхідні значення гарантовано є додатніми цілими числами. Якщо равлик не може дістатися до вершини дерева, має виводитися повідомлення: Impossible

Приклад:

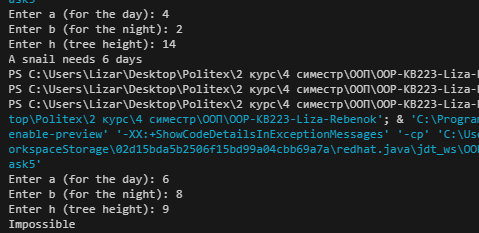
Введення даних: 4 2 14

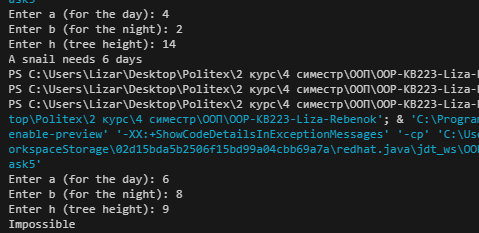
Результат: 6

**Хід виконання завдання:**

****

**Вивід:**

****

****

**Текст програми:**

|  |
| --- |
| import java.util.Scanner;    public class Task5 {    public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.print("Enter a (for the day): ");  int a = scanner.nextInt();  System.out.print("Enter b (for the night): ");  int b = scanner.nextInt();  System.out.print("Enter h (tree height): ");  int h = scanner.nextInt();  if (a <= b) {  System.out.println("Impossible");  } else {  int days = (h - a) / (a - b);  if ((h - a) % (a - b) != 0) {  days++;  }  days++;  System.out.println("A snail needs " + days + " days");  }  scanner.close();  }    } |

**Завдання №6.** Дано така умова: компанія друзів відвідує ресторан. Вони вирішили розділити рахунок порівну та додати 10 відсотків від загальної суми рахунку як чайові. Далі друзі покривають рівними частинами загальну суму платежу.

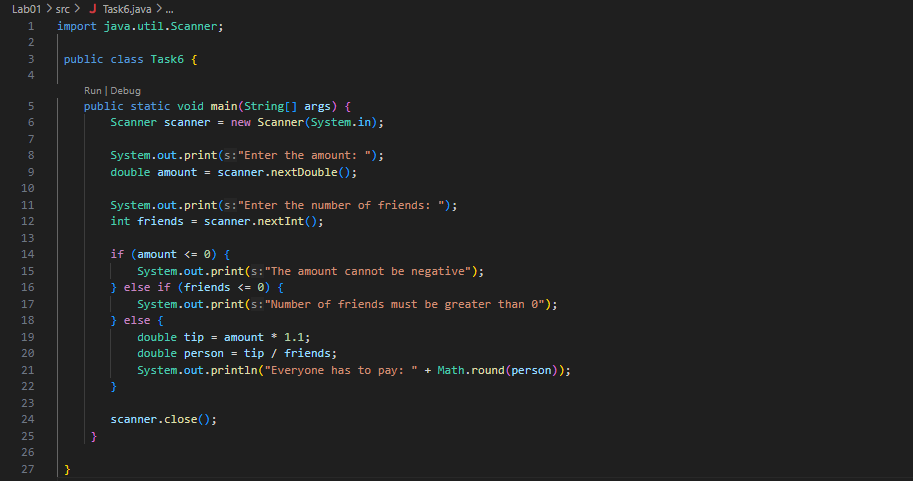
Напишіть програму, яка зчитує загальну суму рахунку та кількість друзів, а потім виводить розмір частини оплати. Доповнити програму перевірками на від’ємну суму рахунку та нульову кількість друзів.

Приклад:

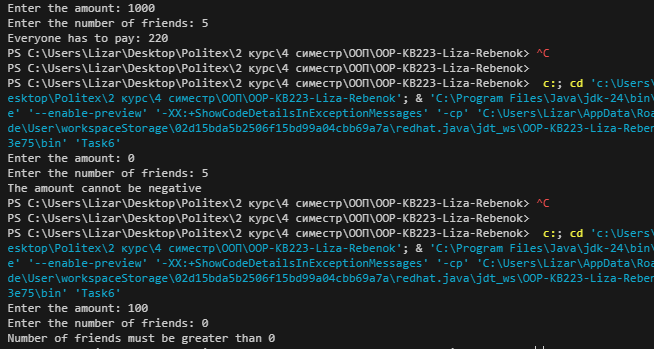
Введення даних: 1000 5

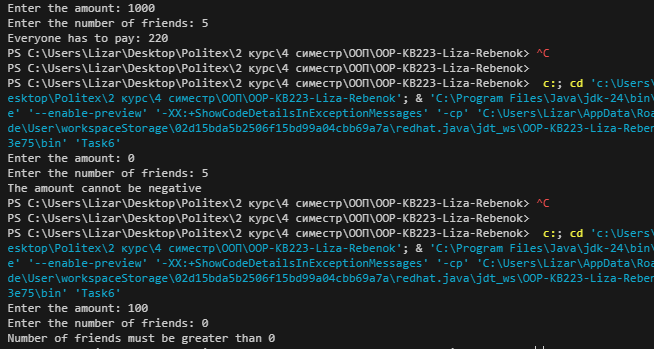
Результат: 220

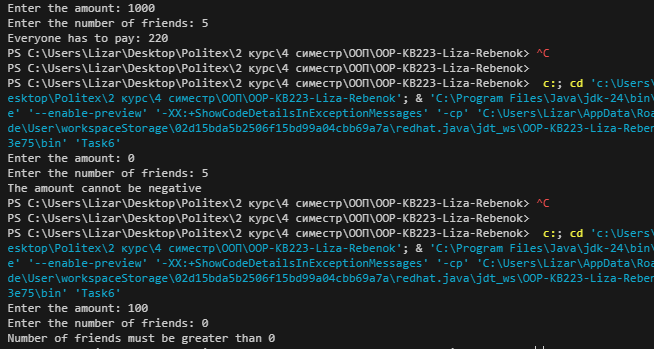
**Хід виконання завдання:**



**Вивід:**

****

****

****

**Текст програми:**

|  |
| --- |
| import java.util.Scanner;    public class Task6 {    public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);    System.out.print("Enter the amount: ");  double amount = scanner.nextDouble();  System.out.print("Enter the number of friends: ");  int friends = scanner.nextInt();  if (amount <= 0) {  System.out.print("The amount cannot be negative");  } else if (friends <= 0) {  System.out.print("Number of friends must be greater than 0");  } else {  double tip = amount \* 1.1;  double person = tip / friends;  System.out.println("Everyone has to pay: " + Math.round(person));  }  scanner.close();  }    } |

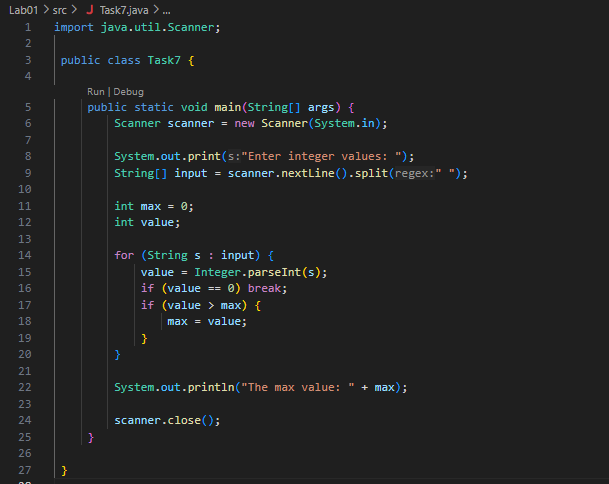
**Завдання №7.** Напишіть програму, яка зчитує послідовність цілих значень зі стандартного введення і знаходить максимальне з них. Ви повинні зчитувати значення послідовності, доки наступне не стане 0. Нульове значення означає кінець вхідної послідовності та не є її елементом. Послідовність гарантовано містить хоча б одне значення.

Приклад:

Введення даних: 2 4 6 9 2 4 5 0

Результат: 9

**Хід виконання завдання:**

****

**Вивід:**

****

**Текст програми:**

|  |
| --- |
| import java.util.Scanner;    public class Task7 {    public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);    System.out.print("Enter integer values: ");  String[] input = scanner.nextLine().split(" ");    int max = 0;  int value;    for (String s : input) {  value = Integer.parseInt(s);  if (value == 0) break;  if (value > max) {  max = value;  }  }    System.out.println("The max value: " + max);    scanner.close();  }    } |

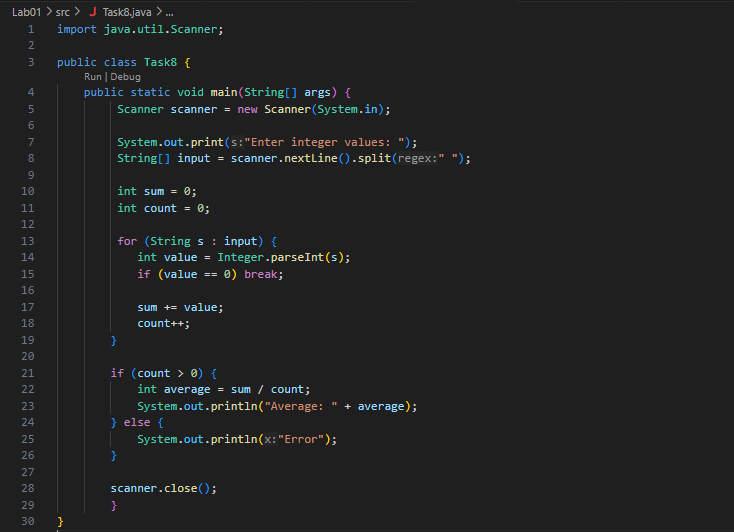
**Завдання №8.** Напишіть програму, яка зчитує послідовність цілих значень зі стандартного введення та знаходить середнє значення. Ви повинні зчитувати значення послідовності, доки наступне не стане 0. Нульове значення означає кінець вхідної послідовності та не є її частиною. Послідовність гарантовано містить хоча б одне значення.

Приклад:

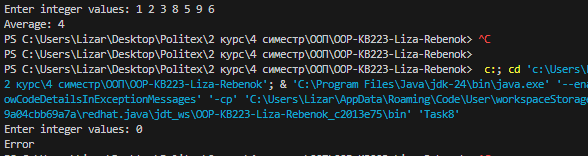
Введення даних: 1 2 3 4 5 0

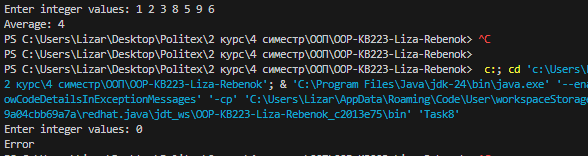
Результат: 3

**Хід виконання завдання:**

****

**Вивід:**

****

****

**Текст програми:**

|  |
| --- |
| import java.util.Scanner;  public class Task8 {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);    System.out.print("Enter integer values: ");  String[] input = scanner.nextLine().split(" ");    int sum = 0;  int count = 0;    for (String s : input) {  int value = Integer.parseInt(s);  if (value == 0) break;  sum += value;  count++;  }  if (count > 0) {  int average = sum / count;  System.out.println("Average: " + average);  } else {  System.out.println("Error");  }  scanner.close();  }  } |

**Завдання №9.** Напишіть програму для вирішення квадратних рівнянь. При виконанні врахуйте залежність значення дискримінанту до кількості результатів.

Приклад:

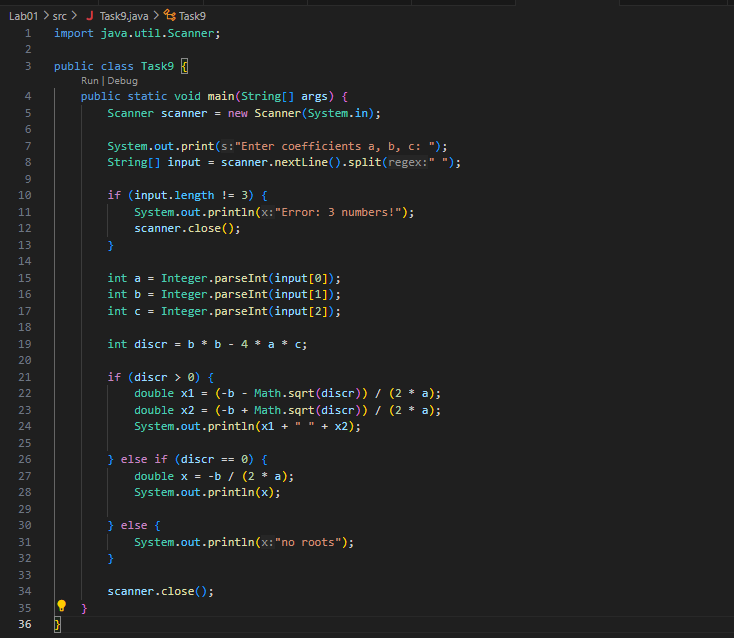
Введення даних: 2 5 -3

Результат: -3 0.5

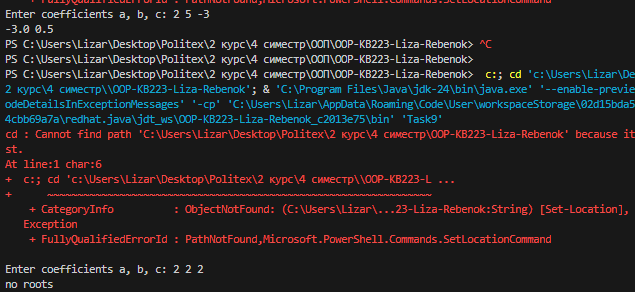
Введення даних: 2 2 2

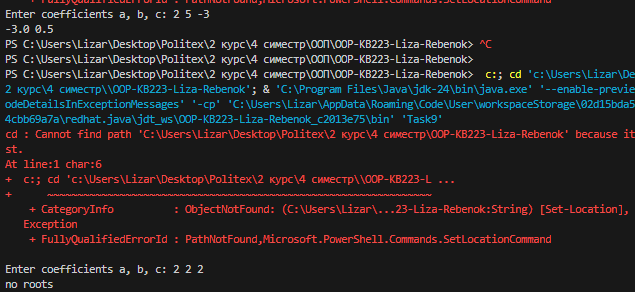
Результат : no roots

**Хід виконання завдання:**

****

**Вивід:**

****

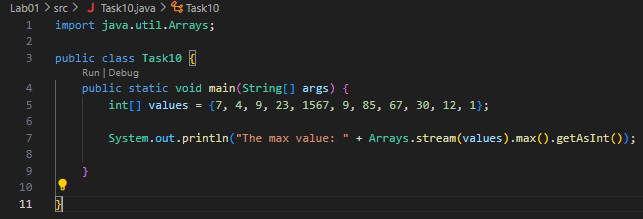
****

**Текст програми:**

|  |
| --- |
| import java.util.Scanner;  public class Task9 {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.print("Enter coefficients a, b, c: ");  String[] input = scanner.nextLine().split(" ");  if (input.length != 3) {  System.out.println("Error: 3 numbers!");  scanner.close();  }  int a = Integer.parseInt(input[0]);  int b = Integer.parseInt(input[1]);  int c = Integer.parseInt(input[2]);  int discr = b \* b - 4 \* a \* c;  if (discr > 0) {  double x1 = (-b - Math.sqrt(discr)) / (2 \* a);  double x2 = (-b + Math.sqrt(discr)) / (2 \* a);  System.out.println(x1 + " " + x2);  } else if (discr == 0) {  double x = -b / (2 \* a);  System.out.println(x);  } else {  System.out.println("no roots");  }  scanner.close();  }  } |

**Завдання №10.** Напишіть програму, що містить метод max(…). Правильна реалізація має отримати масив значень типу int та повернути його максимальне значення. Вхідний масив гарантовано не буде порожнім або null. Метод max не повинен змінювати масив.

**Хід виконання завдання:**

****

**Вивід:**

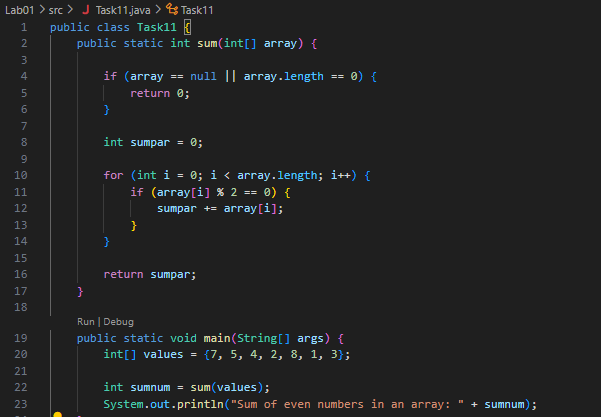
****

**Текст програми:**

|  |
| --- |
| import java.util.Arrays;  public class Task10 {  public static void main(String[] args) {  int[] values = {7, 4, 9, 23, 1567, 9, 85, 67, 30, 12, 1};  System.out.println("The max value: " + Arrays.stream(values).max().getAsInt());  }  } |

**Завдання №11.** Напишіть програму, що містить метод sum. Правильна реалізація має отримати масив значень int та повернути суму парних чисел. Якщо цей масив дорівнює нулю або порожній, метод повертає 0. Метод sum не повинен змінювати масив.

**Хід виконання завдання:**

****

**Вивід:**

****

**Текст програми:**

|  |
| --- |
| public class Task11 {  public static int sum(int[] array) {  if (array == null || array.length == 0) {  return 0;  }    int sumpar = 0;  for (int i = 0; i < array.length; i++) {  if (array[i] % 2 == 0) {  sumpar += array[i];  }  }    return sumpar;  }    public static void main(String[] args) {  int[] values = {7, 5, 4, 2, 8, 1, 3};    int sumnum = sum(values);  System.out.println("Sum of even numbers in an array: " + sumnum);  }  } |

**Завдання №12.** Напишіть програму, що містить метод getSumCheckArray. Правильна реалізація повинна отримати масив значень int і повернути масив логічних значень, де кожен елемент є результатом перевірки: чи є вихідний елемент сумою двох попередніх елементів у даному масиві.

Гарантується, що довжина масиву становить 2 або більше. Цей масив гарантовано не дорівнює null. Метод повертає масив логічних значень, де кожен елемент є результатом для відповідного елемента в даному масиві. Перші два елементи логічного масиву завжди false.

Приклад

Введення даних: [1, -1, 0, 4, 6, 10, 15, 25]

Результат: [false, false, true, false, false, true, false, true]

**Хід виконання завдання:**

****

**Вивід:**

****

**Текст програми:**

|  |
| --- |
| import java.util.Arrays;    public class Task12 {    public static void main(String[] args) {  int[] values = {1, -1, 0, 4, 6, 10, 15, 25};    System.out.println(Arrays.toString(getSumCheckArray(values)));    }    private static boolean[] getSumCheckArray(int[] values) {  boolean[] result = new boolean[values.length];    for (int i = 2; i < values.length; ++i) {  if (values[i-2] + values[i-1] == values[i]) {  result[i] = true;  }  else {  result[i] = false;  }  }    return result;  }    } |

**Завдання №13.** Напишіть програму, що містить метод removeLocalMaxima(…). Правильна реалізація має отримати масив значень типу int і повернути копію даного масиву з усіма видаленими локальними максимумами. Вихідний масив не можна змінювати.

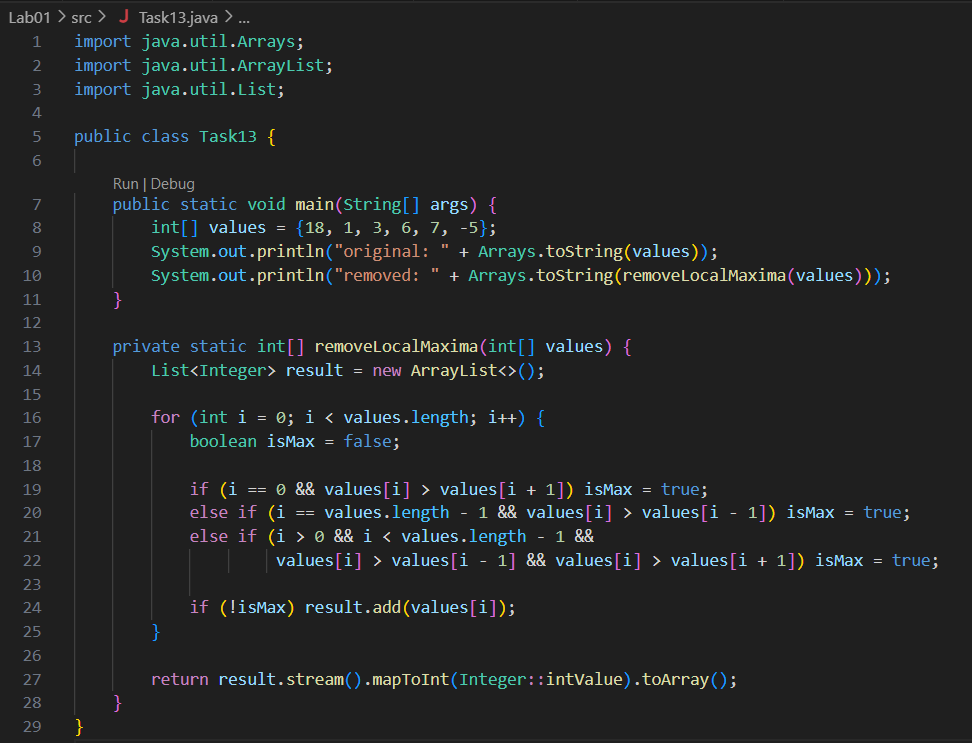
Локальний максимум — це елемент, який більший за будь-який з його сусідніх елементів. Вам необхідно видалити елементи, які є локальними максимумами у вихідному масиві. Розмір цього масиву гарантовано більше 1. Якщо масив не має локальних максимумів, ви повинні повернути його копію без змін. Ви можете використовувати методи java.util.Arrays.\*.

Приклад:

Введення даних: [18, 1, 3, 6, 7, -5]

Результат виконання: [1, 3, 6, -5]

**Хід виконання завдання:**

****

**Вивід:**

****

**Текст програми:**

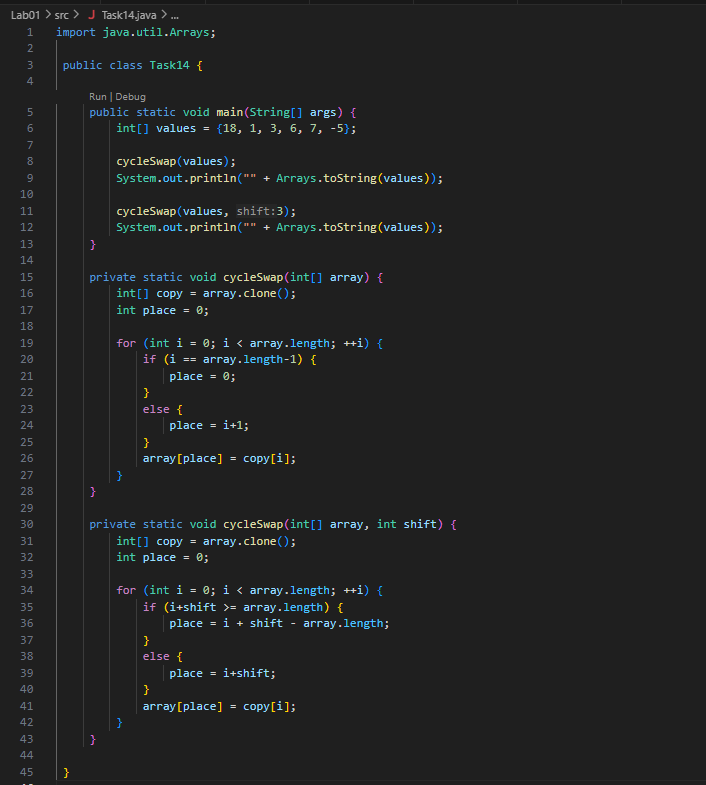
|  |
| --- |
| import java.util.Arrays;  import java.util.ArrayList;  import java.util.List;  public class Task13 {  public static void main(String[] args) {  int[] values = {18, 1, 3, 6, 7, -5};  System.out.println("original: " + Arrays.toString(values));  System.out.println("removed: " + Arrays.toString(removeLocalMaxima(values)));  }  private static int[] removeLocalMaxima(int[] values) {  List<Integer> result = new ArrayList<>();  for (int i = 0; i < values.length; i++) {  boolean isMax = false;  if (i == 0 && values[i] > values[i + 1]) isMax = true;  else if (i == values.length - 1 && values[i] > values[i - 1]) isMax = true;  else if (i > 0 && i < values.length - 1 &&  values[i] > values[i - 1] && values[i] > values[i + 1]) isMax = true;  if (!isMax) result.add(values[i]);  }  return result.stream().mapToInt(Integer::intValue).toArray();  }  } |

**Завдання №14.** Напишіть програму що містить два методи, що реалізують наступний функціонал:

void cycleSwap(int[] array) - Зсуває всі елементи в даному масиві вправо на 1 позицію. В цьому випадку останній елемент масиву стає першим. Наприклад, 1 3 2 7 4 стає 4 1 3 2 7.

void cycleSwap(int[] array, int shift) - Зсуває всі елементи в заданому масиві вправо на shift позицій. Гарантується, що значення зсуву невід'ємне і не більше за довжину масиву. Наприклад, 1 3 2 7 4 зі зсувом 3 стає 2 7 4 1 3.

**Хід виконання завдання:**

****

**Вивід:**

****

**Текст програми:**

|  |
| --- |
| import java.util.Arrays;    public class Task14 {    public static void main(String[] args) {  int[] values = {18, 1, 3, 6, 7, -5};    cycleSwap(values);  System.out.println("" + Arrays.toString(values));    cycleSwap(values, 3);  System.out.println("" + Arrays.toString(values));  }    private static void cycleSwap(int[] array) {  int[] copy = array.clone();  int place = 0;    for (int i = 0; i < array.length; ++i) {  if (i == array.length-1) {  place = 0;  }  else {  place = i+1;  }  array[place] = copy[i];  }  }    private static void cycleSwap(int[] array, int shift) {  int[] copy = array.clone();  int place = 0;    for (int i = 0; i < array.length; ++i) {  if (i+shift >= array.length) {  place = i + shift - array.length;  }  else {  place = i+shift;  }  array[place] = copy[i];  }  }    } |

Посилання на github: <https://github.com/Lizarebenokk/OOP-KB-232-Liza-Rebenok/tree/main>