# 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 до лабораторної роботи

Встановити JDK на власному персональному комп’ютері та будь-яку IDE. Також допускається виконання лабораторних робіт у будь-якому іншому текстовому редакторі.

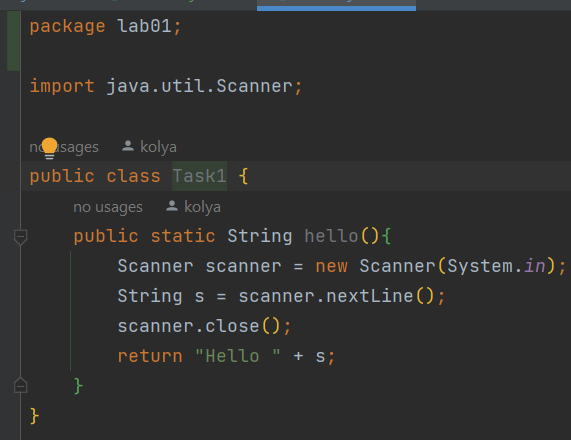
Створити акаунт на github у випадку відсутності.

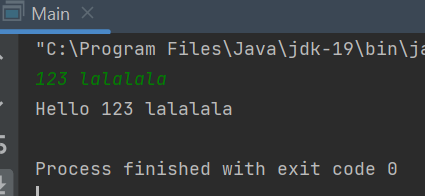
Створити репозиторій що має назву яка задовольняє наступному шаблону **OOP-KB-22{1 or 2}-Name-Surname** (наприклад OOP-KB223-Ihor-Diuba). Результати роботи необхідно буде завантажити на github для перевірки.

При використанні IDE створити проект під назвою **lab01** для розміщення результатів виконання завдань. У випадку використання звичайного текстового редактору створити окрему директорію **lab01** для виконання завдань.

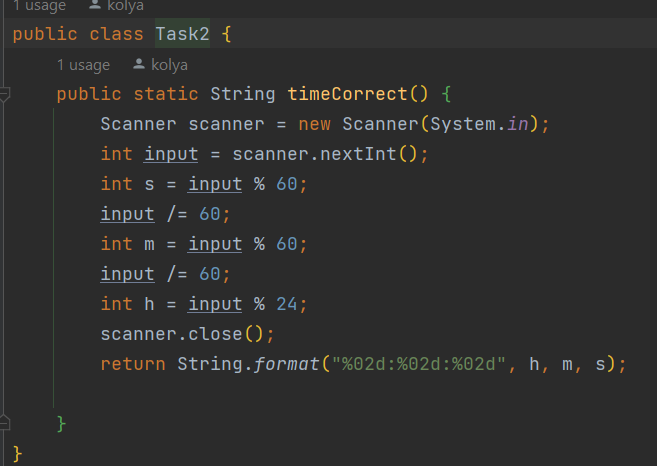
Виконайте наступні завдання(кожне завдання має бути виконано в окремому файлі):

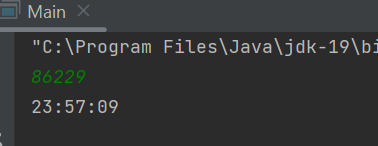
|  |  |
| --- | --- |
| 1 | Напишіть програму, яка прочитає рядок із System.in і виведе повідомлення: "Hello, прочитаний рядок". Якщо у вхідному рядку кілька слів, потрібно надрукувати їх усі. |



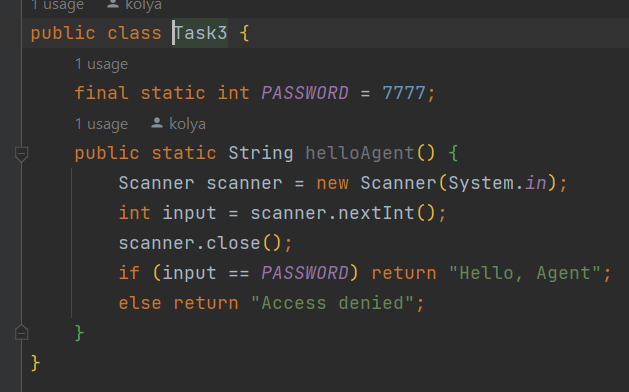


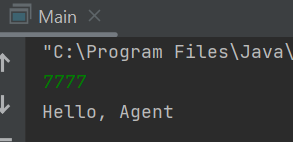
|  |  |
| --- | --- |
| 2 | Напишіть програму, яка отримує на вхід ціле число. Це число є поточний добовий час, який обчислюється як кількість секунд, що пройшли з півночі. Потім програма повинна надрукувати значення екрану електронного годинника для цього часу.  Вхідне значення отримується через System.in. 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]).  **Приклади**:  Введення даних: 60  Результат виконання: 0:01:00  Введення даних: 3599  Результат виконання: 0:59:59  Введення даних: 86229  Результат виконання: 23:57:09 |



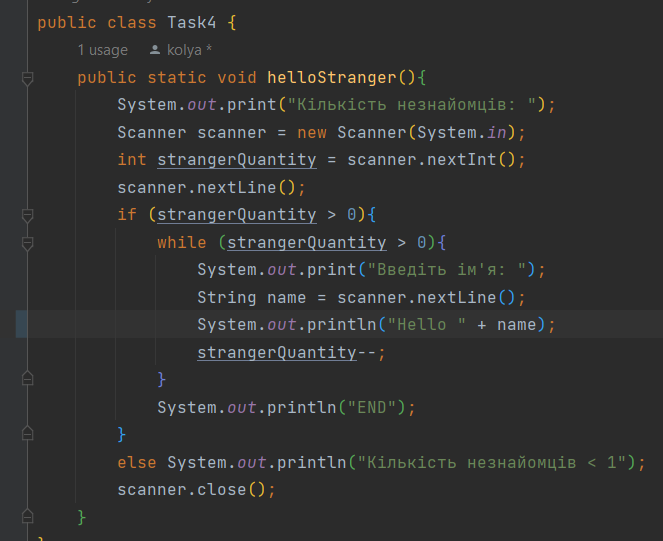


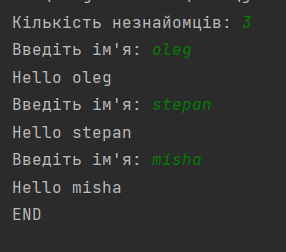
|  |  |
| --- | --- |
| 3 | Напишіть програму, яка запитує на вхід число-пароль. Якщо воно дорівнює секретній кількості, програма виводить: "Hello, Agent", інакше друкує: "Access denied".  Секретний пароль зберігається в final static int password. |



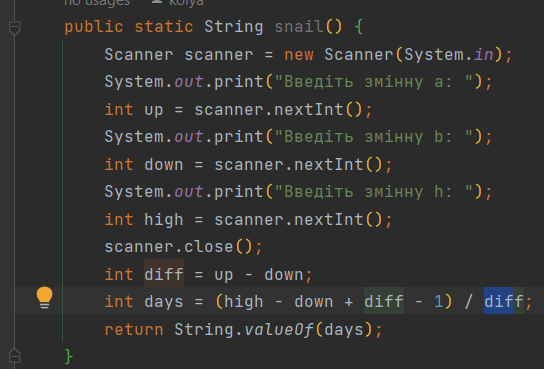


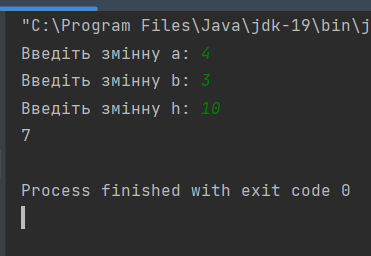
|  |  |
| --- | --- |
| 4 | Напишіть просту програму, яка запитує кількість незнайомців, з якими потрібно зустрітися, рядково читає імена незнайомців, по рядках виводить: "Hello, (ім'я незнайомця)" для кожного незнайомця.  Доповнити програму перевірками на кількість, що є від’ємною, та рівна 0. |



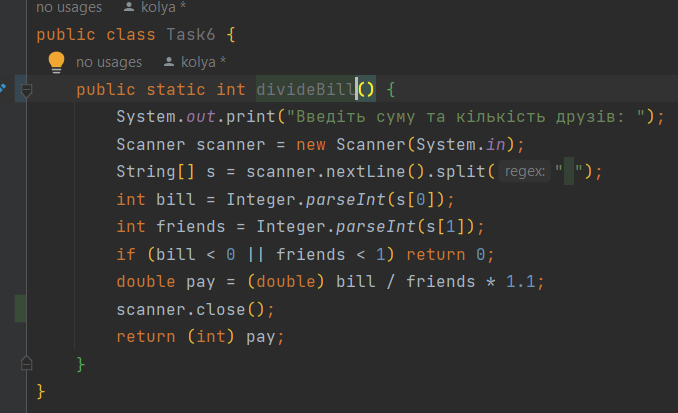


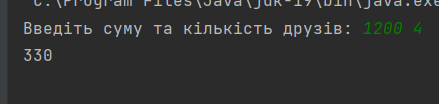
|  |  |
| --- | --- |
| 5 | Наведено таку умову: равлик піднімається по дереву на a футів на день. Потім щоночі равлик сповзає вниз на b футів. Висота дерева — h футів.  Напишіть програму з підрахунком кількості днів, які знадобляться равлику, щоб дістатися до вершини дерева.  Програма читає a, b, h рядково. Вхідні значення гарантовано є додатніми цілими числами.  Якщо равлик не може дістатися до вершини дерева, має виводитися повідомлення: Impossible  **Приклади**:  Введення даних: 4 2 14  Результат: 6  Введення даних: 4 3 10  Результат: 7  Введення даних: 4 4 10  Результат: Impossible |



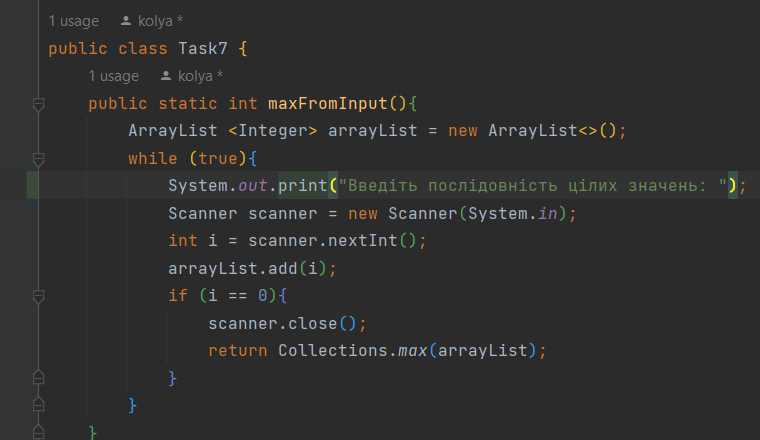


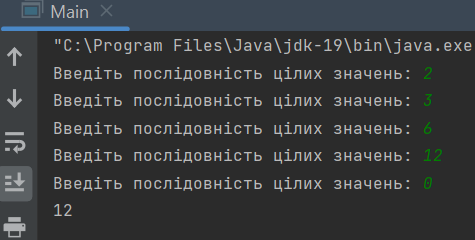
|  |  |
| --- | --- |
| 6 | Дано така умова: компанія друзів відвідує ресторан. Вони вирішили розділити рахунок порівну та додати 10 відсотків від загальної суми рахунку як чайові. Далі друзі покривають рівними частинами загальну суму платежу.  Напишіть програму, яка зчитує загальну суму рахунку та кількість 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 |



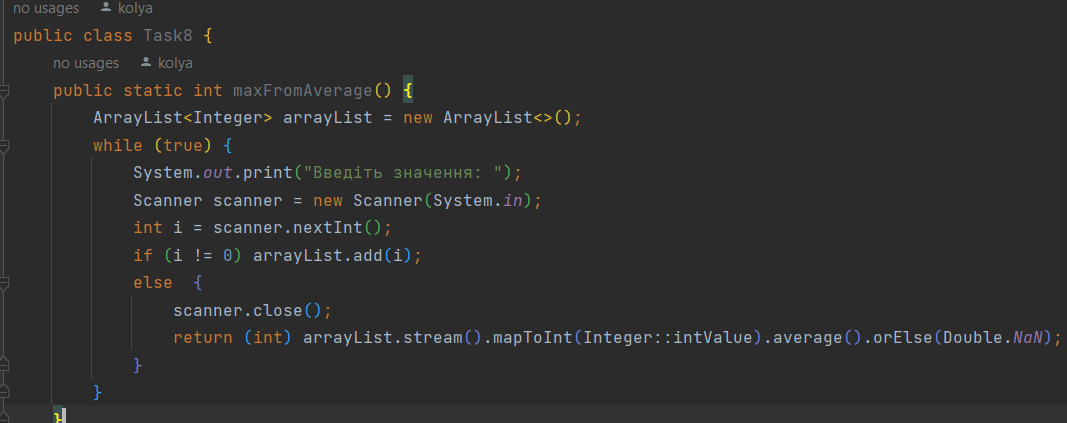


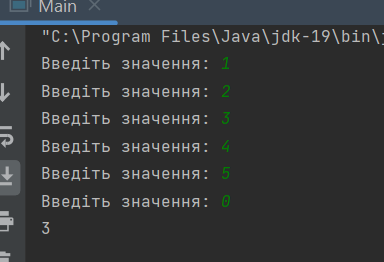
|  |  |
| --- | --- |
| 7 | Напишіть програму, яка зчитує послідовність цілих значень зі стандартного введення і знаходить максимальне з них. Ви повинні зчитувати значення послідовності, доки наступне не стане 0. Нульове значення означає кінець вхідної послідовності та не є її елементом. Послідовність гарантовано містить хоча б одне значення.  **Приклад**:  Введення даних: 2 4 6 9 2 4 5 0  Результат: 9 |



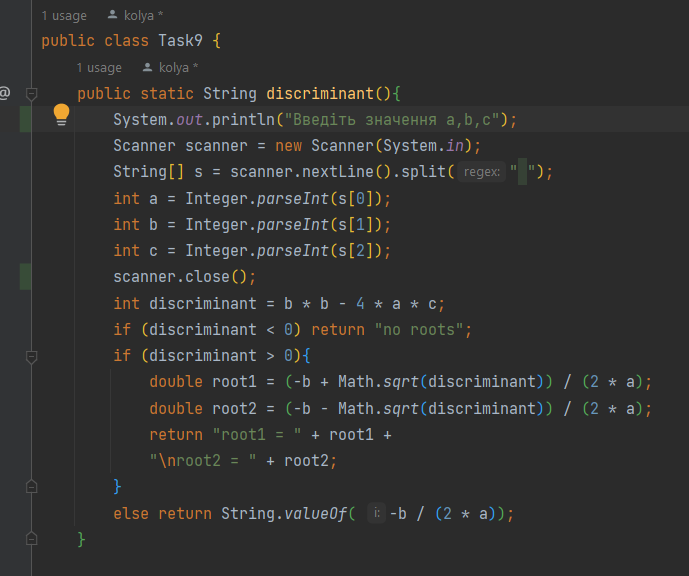


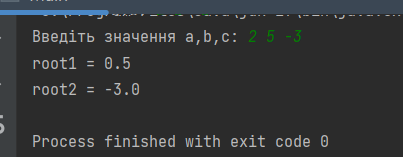
|  |  |
| --- | --- |
| 8 | Напишіть програму, яка зчитує послідовність цілих значень зі стандартного введення та знаходить середнє значення. Ви повинні зчитувати значення послідовності, доки наступне не стане 0. Нульове значення означає кінець вхідної послідовності та не є її частиною. Послідовність гарантовано містить хоча б одне значення.  **Приклад:**  Введення даних: 1 2 3 4 5 0  Результат: 3 |



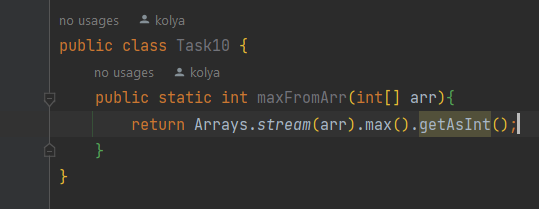


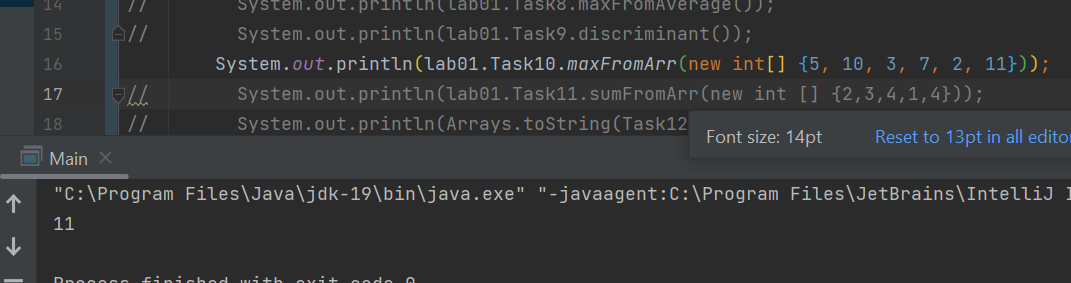
|  |  |
| --- | --- |
| 9 | Напишіть програму для вирішення квадратних рівнянь. При виконанні врахуйте залежність значення дискримінанту до кількості результатів  Приклад:  Введення даних: 2 5 -3  Результат: -3 0.5  Введення даних: 2 2 2  Результат : no roots |



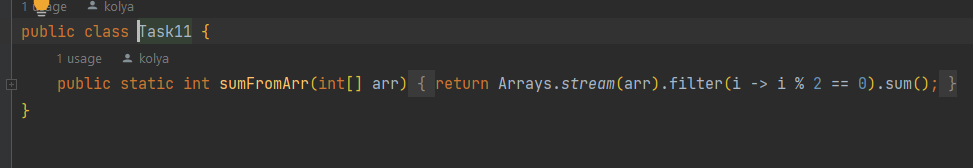


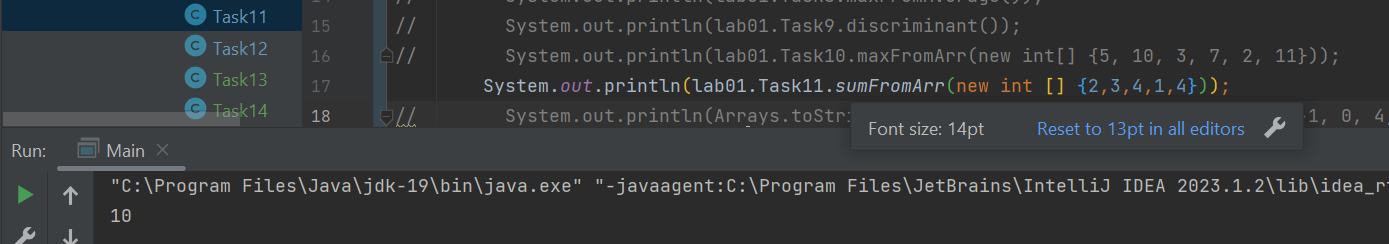
|  |  |
| --- | --- |
| 10 | Напишіть програму, що містить метод max(…). Правильна реалізація має отримати масив значень типу int та повернути його максимальне значення. Вхідний масив гарантовано не буде порожнім або null. Метод max не повинен змінювати масив. |



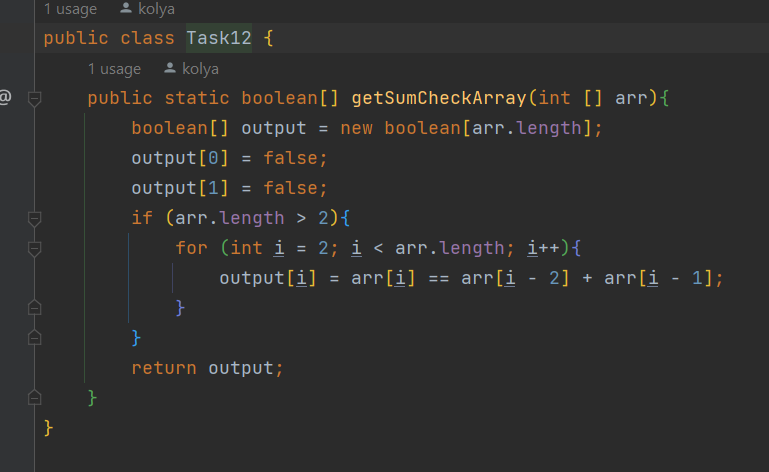


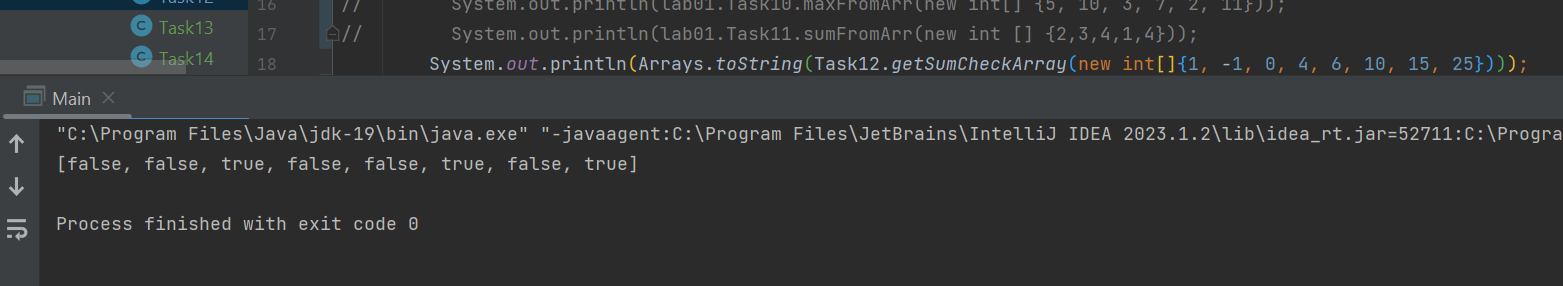
|  |  |
| --- | --- |
| 11 | Напишіть програму, що містить метод sum. Правильна реалізація має отримати масив значень int та повернути суму парних чисел. Якщо цей масив дорівнює нулю або порожній, метод повертає 0. Метод sum не повинен змінювати масив. |



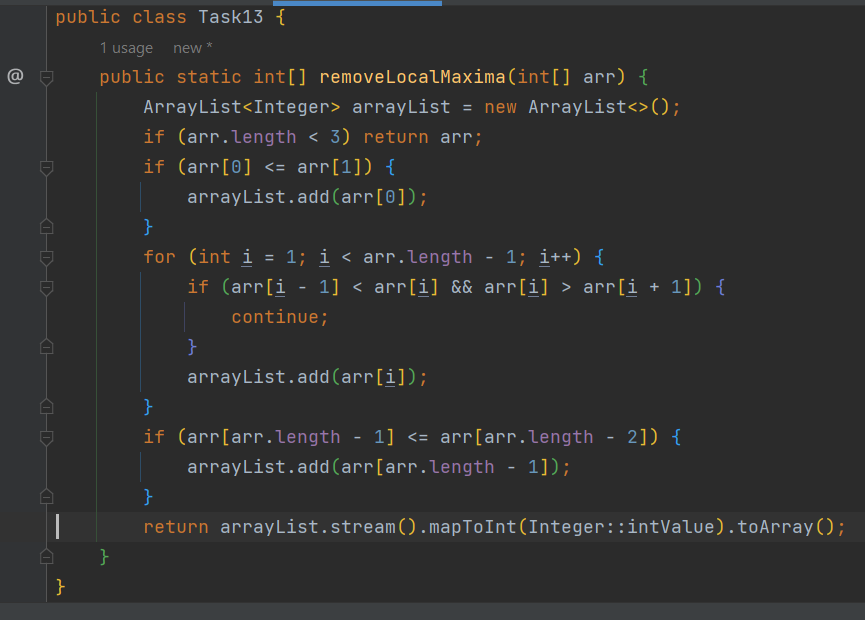


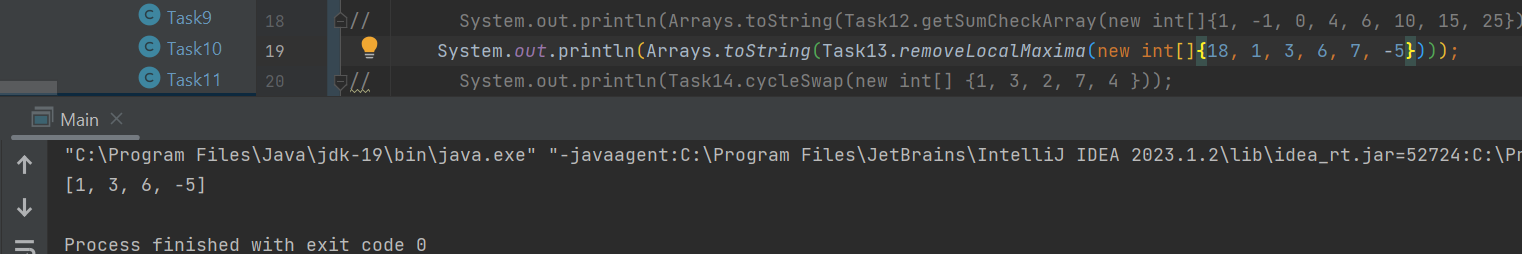
|  |  |
| --- | --- |
| 12 | Напишіть програму, що містить метод getSumCheckArray. Правильна реалізація повинна отримати масив значень int і повернути масив логічних значень, де кожен елемент є результатом перевірки: чи є вихідний елемент сумою двох попередніх елементів у даному масиві.  Гарантується, 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] |





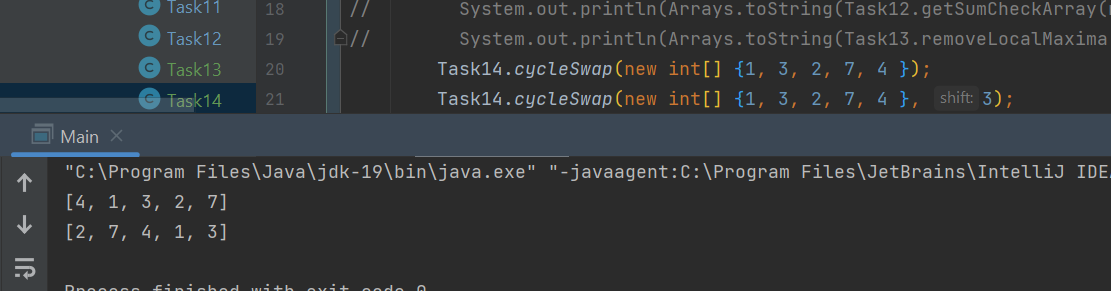
|  |  |
| --- | --- |
| 13 | Напишіть програму, що містить метод removeLocalMaxima(…). Правильна реалізація має отримати масив значень типу int і повернути копію даного масиву з усіма видаленими локальними максимумами. Вихідний масив не можна змінювати.  **Локальний максимум** — це елемент, який більший за будь-який з його сусідніх елементів. Вам необхідно видалити елементи, які є локальними максимумами у вихідному масиві.   * Розмір цього масиву гарантовано більше 1. * 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] |





|  |  |
| --- | --- |
| 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. |





**Висновок:** в ході виконання лабораторної роботи я повторив теоретичні знання та виконав надані завдання.