

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРКАСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ БОГДАНА ХМЕЛЬНИЦЬКОГО  
Факультет обчислювальної техніки, інтелектуальних та управляючих систем  
Кафедра інформаційних технологій

**Звіт**  
**з лабораторної роботи №3**  
**Тема:**  
**Динамічні масиви (одновимірні та зубчасті).**

Виконали:

студенти групи КС – 241:

Пальонний Н. М., Дінов А. І.

студент групи КС – 242:

Ігнатенко В. В.

Перевірив:

старший викладач

Гребенович Ю. Є.

# Постановка завдання

**Тема:** Динамічні масиви (одновимірні та зубчасті).

**Мета:** Ознайомитися з методами обробки динамічних масивів, зокрема зубчастих масивів.

**Опис завдання:** Необхідно створити програму, яка працює з одновимірними та зубчастими масивами. Основна мета — навчитися змінювати розмір масивів і реалізовувати різноманітні операції над ними, такі як додавання, видалення елементів, зсуви тощо.

## Блок 1: Робота з одновимірними масивами

### Масив і зміна його розміру

Реалізувати зміну розміру одномірного масиву одним із двох способів:

- За допомогою `Array.Resize(ref arr)`
- Створити новий масив із використанням `new`, скопіювавши до нього потрібні елементи вручну

### Операції над масивом

Реалізувати окремі методи для таких дій:

- Видалення першого парного елемента
- Видалення останнього від'ємного значення
- Видалення елемента за конкретним значенням
- Вставлення елементів після парних чисел або за іншими правилами

### Структура програми

- Кожна операція має бути оформлена як окрема функція
- Необхідно реалізувати окремі методи для введення, виведення та обробки масиву
- Для всіх методів передбачити блок-схеми та діаграми викликів

### Обмеження

- Забороняється використовувати складні вбудовані методи, окрім `Array.Resize` і `new`
- Усі зміни мають бути виконані вручну, без застосування готових функцій на зразок `Array.ConvertAll`

## Блок 2: Робота із зубчастими масивами

### Операції над зубчастим масивом

- Трансформація структури масиву з мінімальним копіюванням даних

- Робота з рядками: додавання, видалення, зокрема за умовами (наприклад, прибрати рядок із максимальним значенням у всьому масиві)

## **Структура реалізації**

- Усі дії над масивом мають бути реалізовані як окремі методи
- Забезпечити можливість змінювати кількість рядків, додавати нові порожні рядки або видаляти за заданими умовами
- Створити діаграму змін посилань, яка ілюструє зміни в структурі масиву

## **Обмеження**

- Можна використовувати допоміжні методи, але основний акцент — на маніпуляціях з посиланнями, а не на переміщенні значень

## **Вимоги до захисту:**

На захисті необхідно пояснити реалізовані підходи до роботи з динамічними масивами, обґрунтувати обрані методи, та показати вміння працювати зі змінами у структурі масивів.

## **Додатково:**

Можна заробити додаткові бали, якщо один із блоків буде реалізовано як частину командного проєкту з використанням спільного репозиторію Git.

# Варіанти виконання

## **Пальонний Назар Михайлович**

*Блок 1* – Варіант 4: Знищити Т елементів, починаючи з номеру К (лише якщо всі такі елементи фактично є в масиві).

*Блок 2* – Варіант 8: Знищити рядок, в якому знаходиться найбільший елемент зубчастого масиву (якщо у різних місцях є кілька елементів з однаковим максимальним значенням, то лише перший з них).

## **Дінов Артем Ігорович:**

*Блок 1* – Варіант 2: Знищити останній від'ємний елемент.

*Блок 2* – Варіант 11: Додати рядок після рядка, що містить максимальний елемент (якщо у різних місцях є кілька елементів з однаковим максимальним значенням, то брати перший з них)

## **Ігнатенко Віктор Васильович:**

*Блок 1* – Варіант 6: Знищити всі елементи з парними індексами

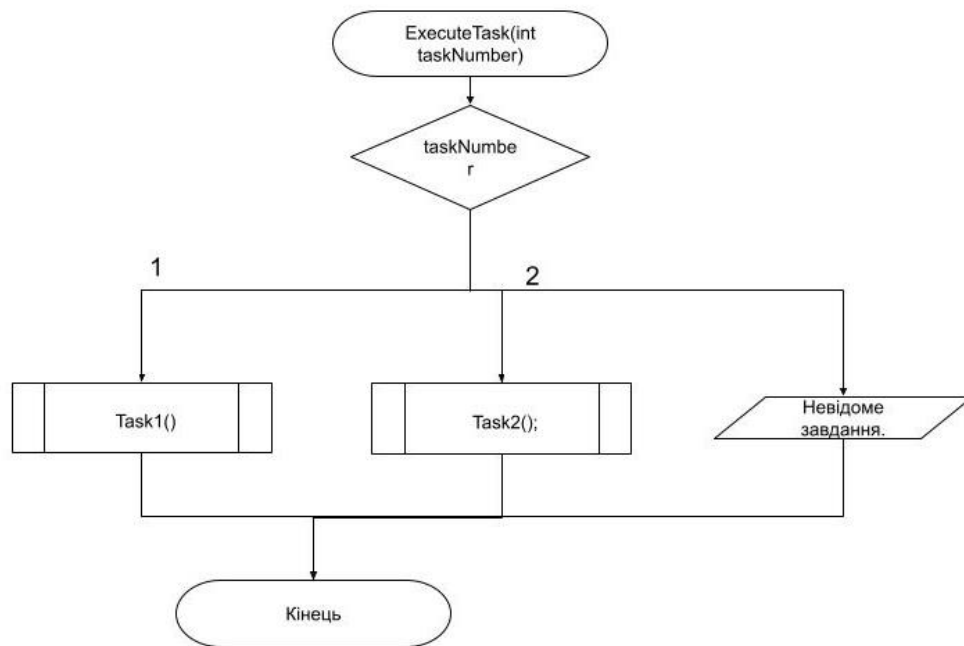
*Блок 2* – Варіант 2: Додати К рядків у початок (угору) зубчастого масиву

## **Лістинг програми:**

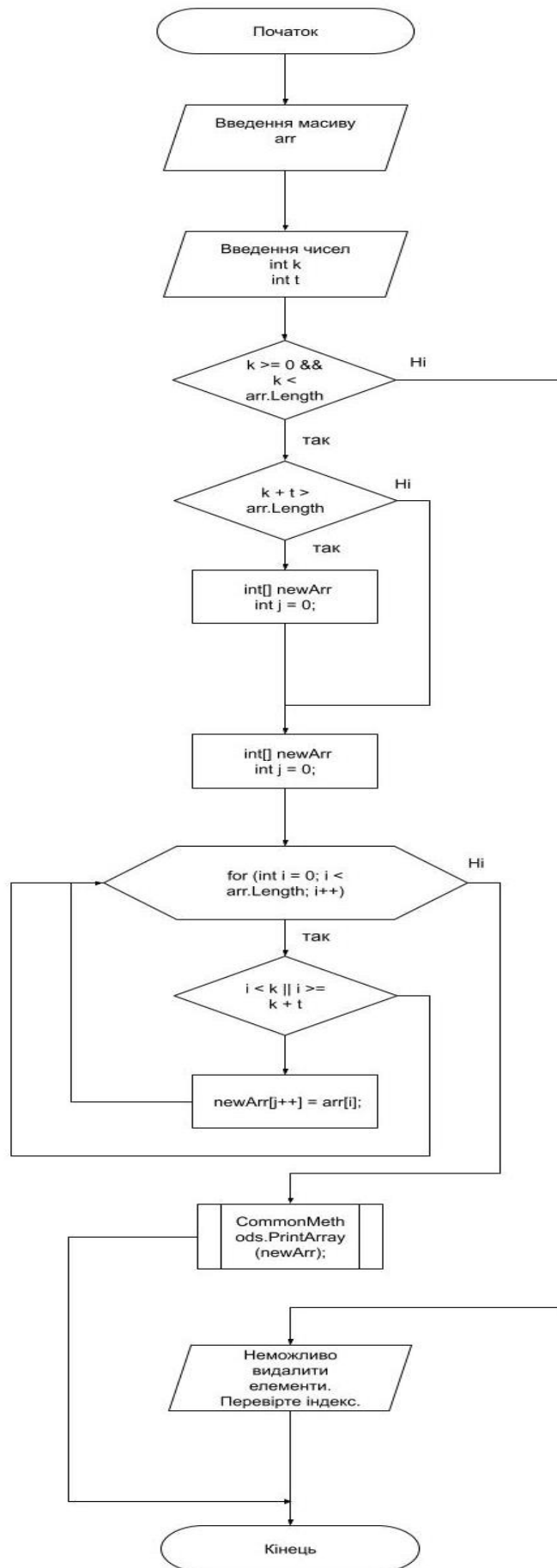
[https://github.com/Artemdin/Lab3\\_Cooperation.git](https://github.com/Artemdin/Lab3_Cooperation.git)

# Блок-схеми

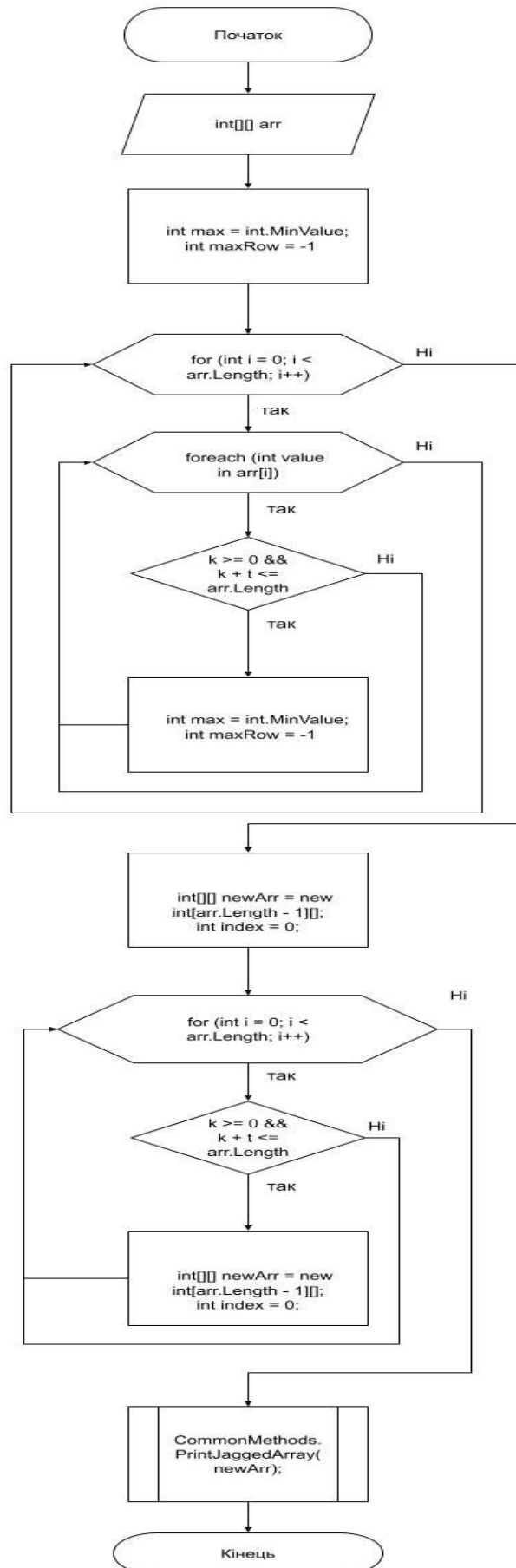
## PalionniySolution



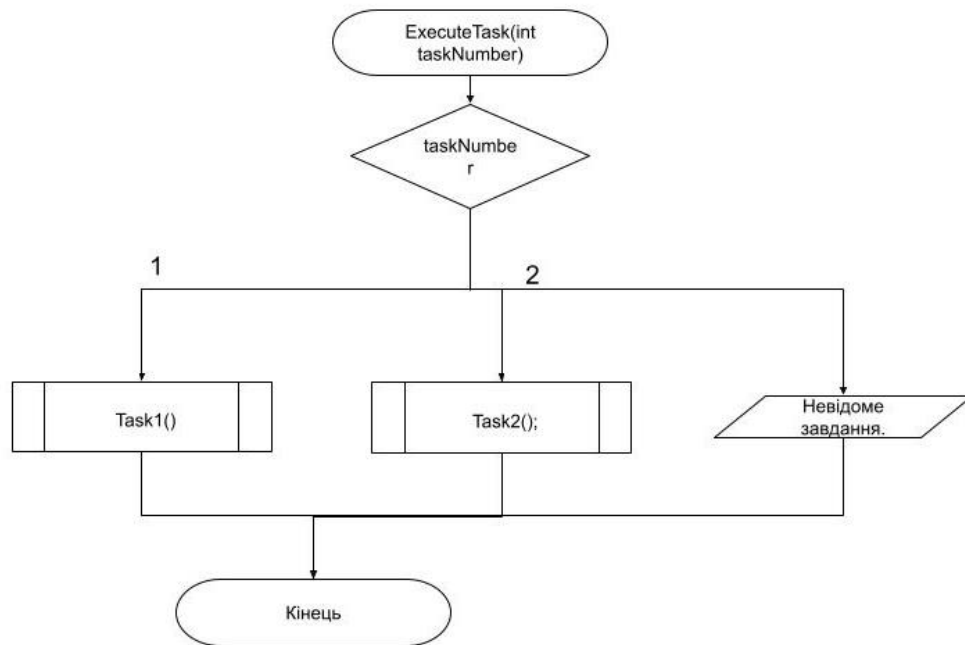
# Task1



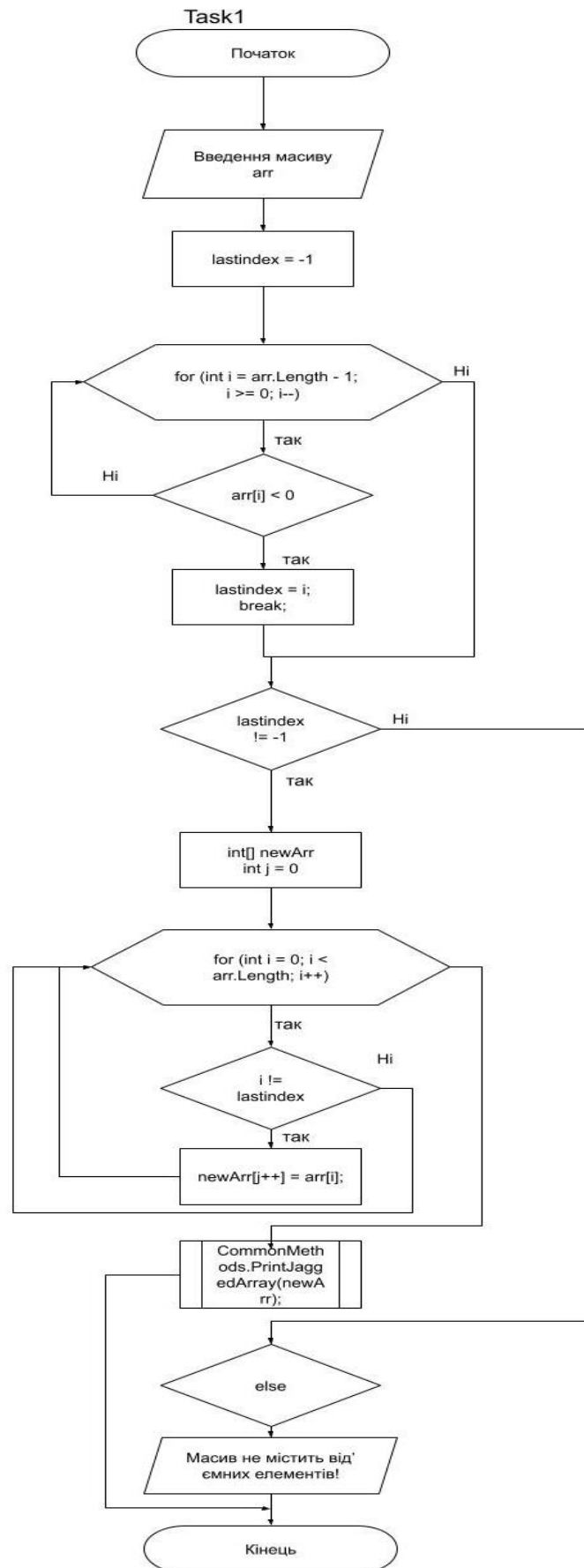
## Task 2



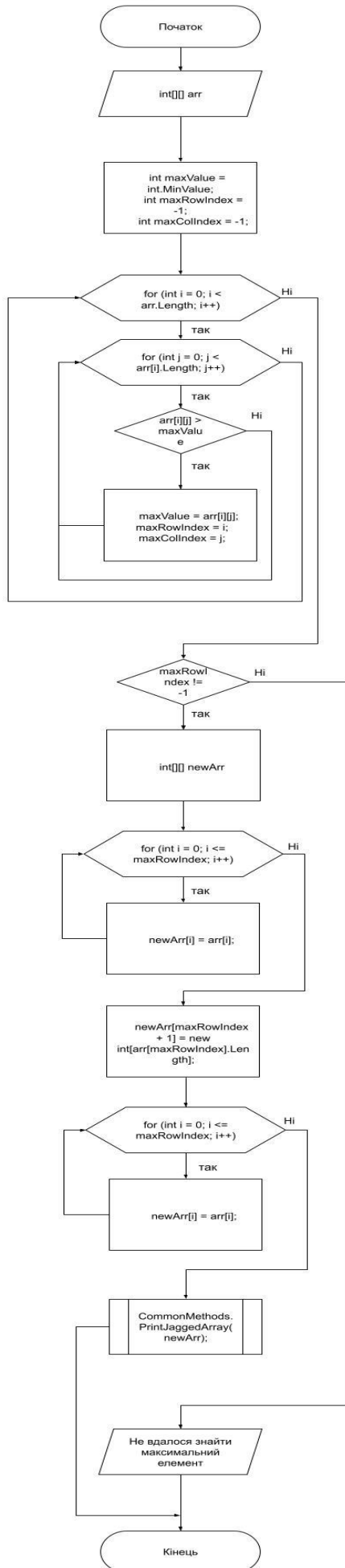
# DinovSolution



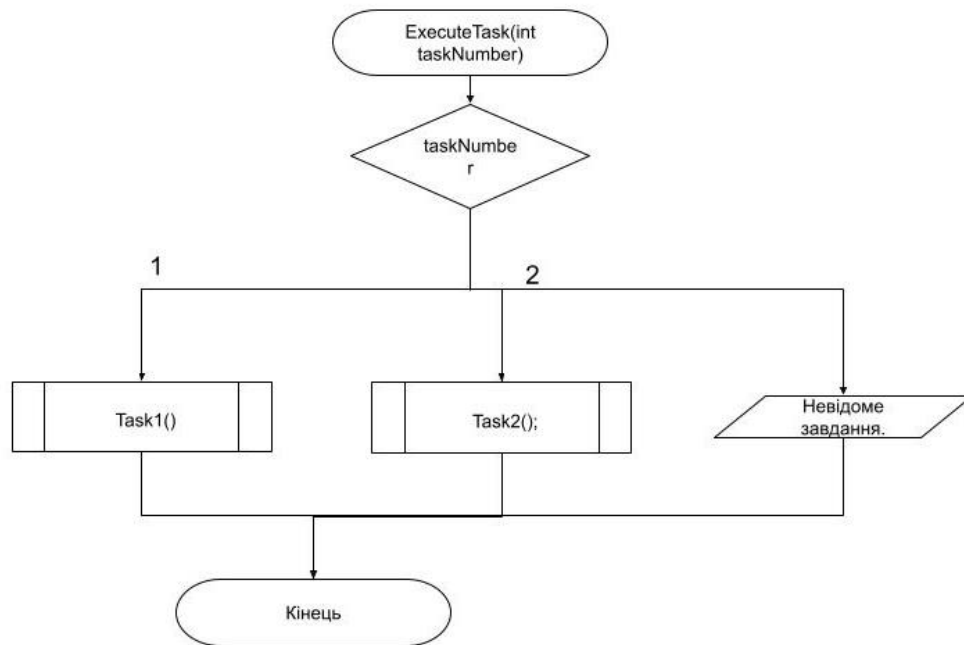




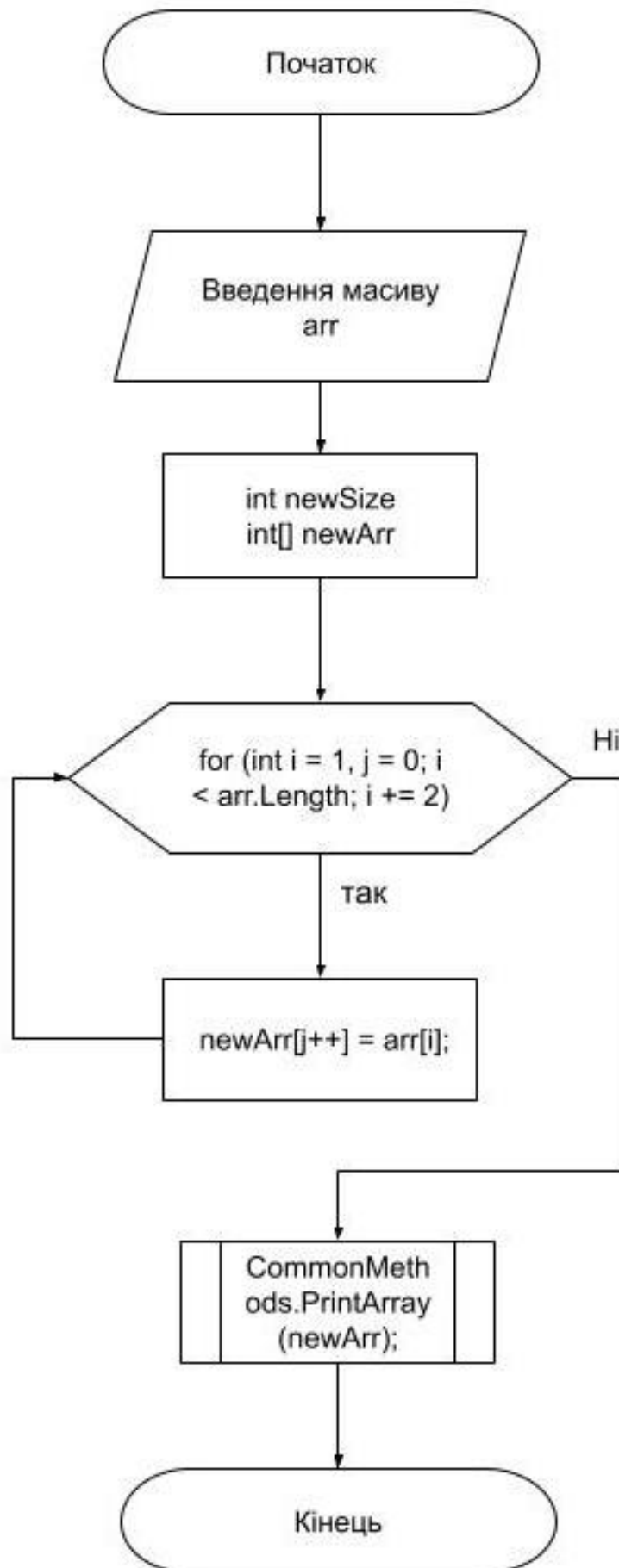
## Task2



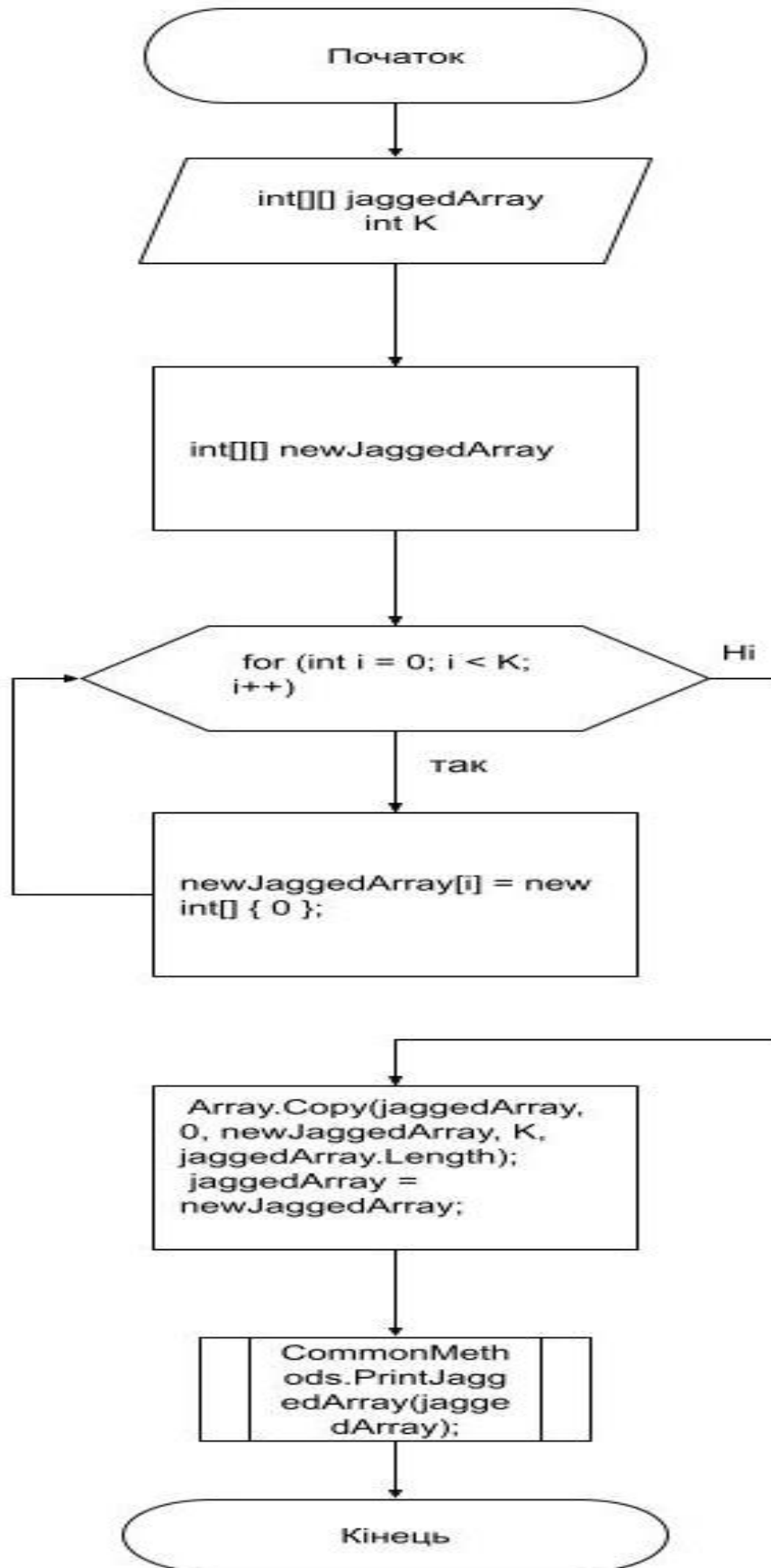
# IgnatenkoSolution



## Task1



## Task2

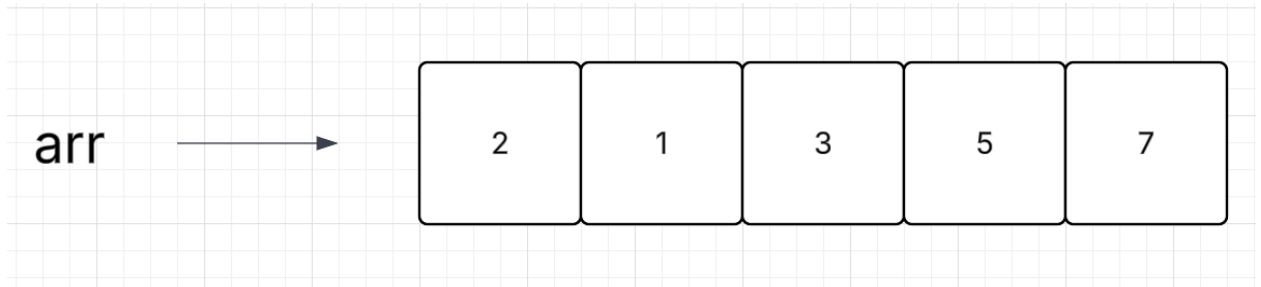


# Діаграми

## Блок 1

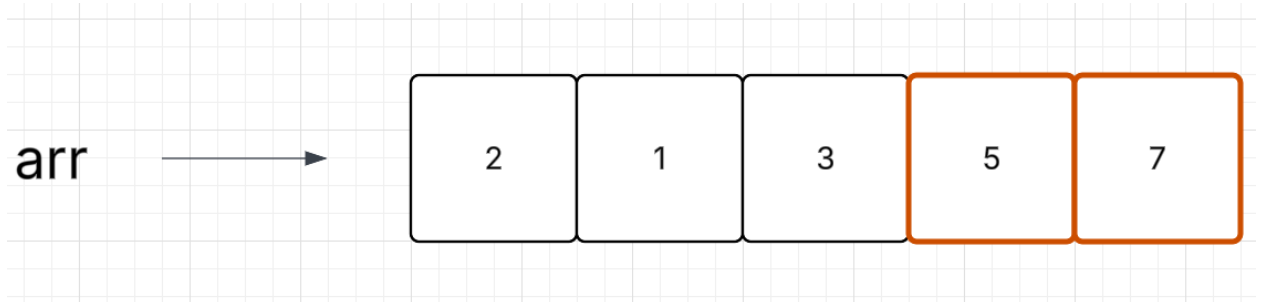
### PalionniySolution.Block\_1\_Task\_4

Створюємо одновимірний масив

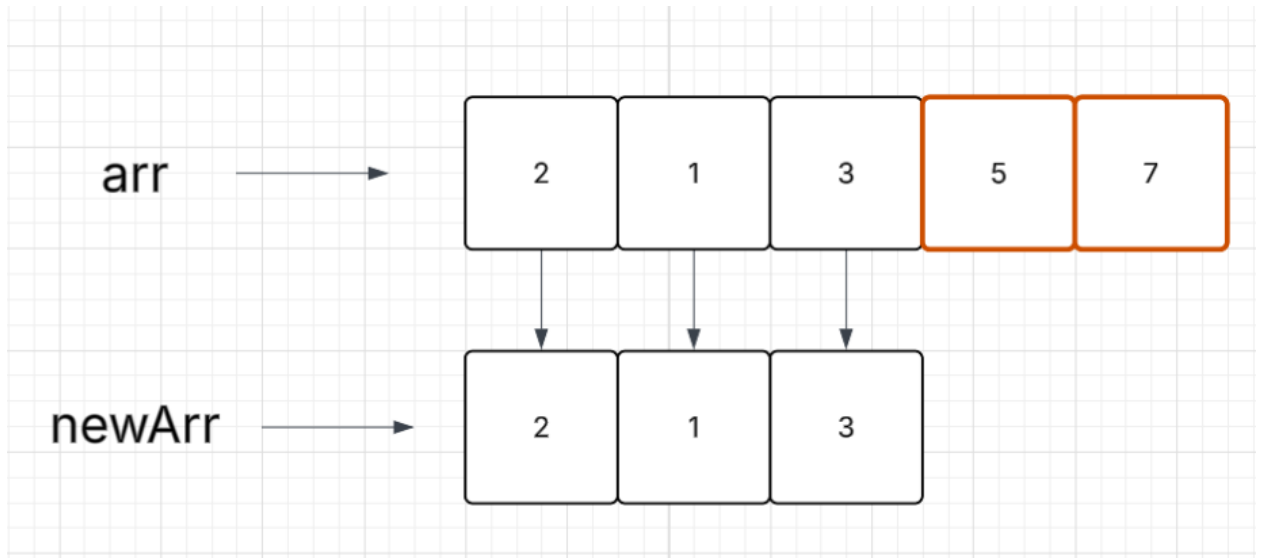


$K=2, t=2$

Знищуємо  $t$  елементів після номеру  $k$

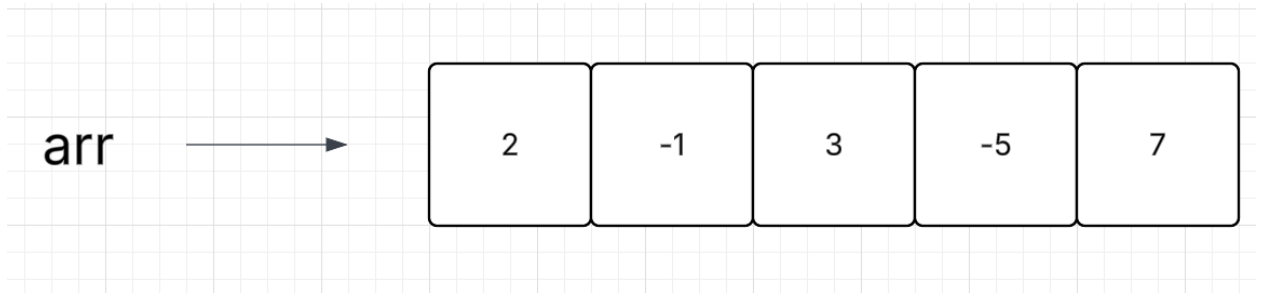


Виводимо результат

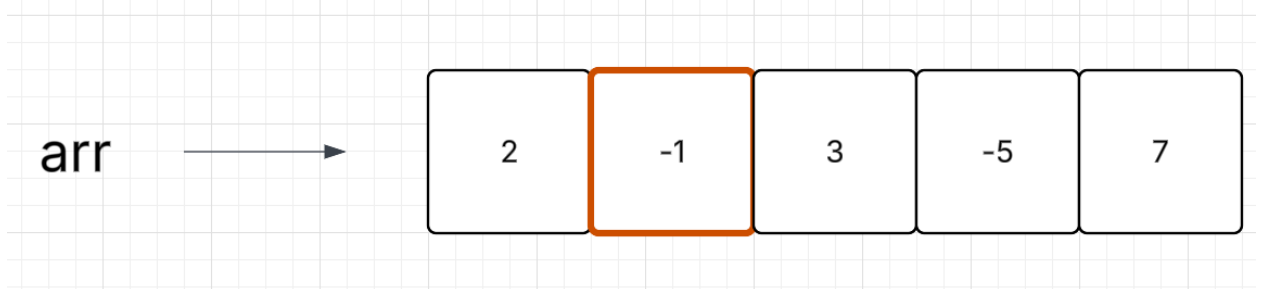


## DinovSolution.Block\_1\_Task\_2

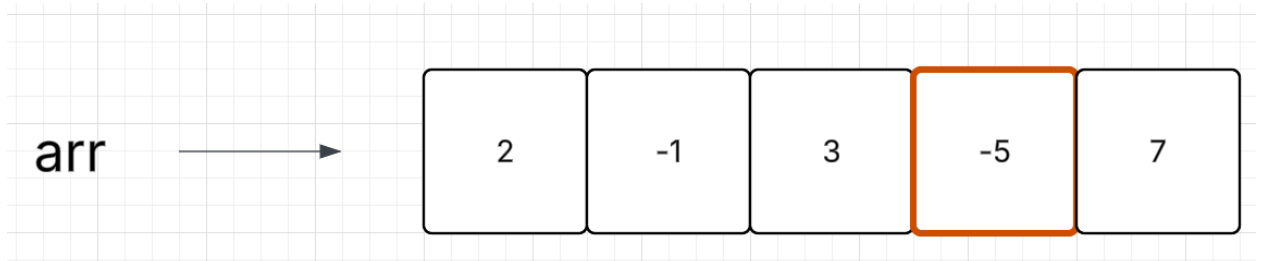
Створюємо одновимірний масив



Знаходимо останній від'ємний елемент

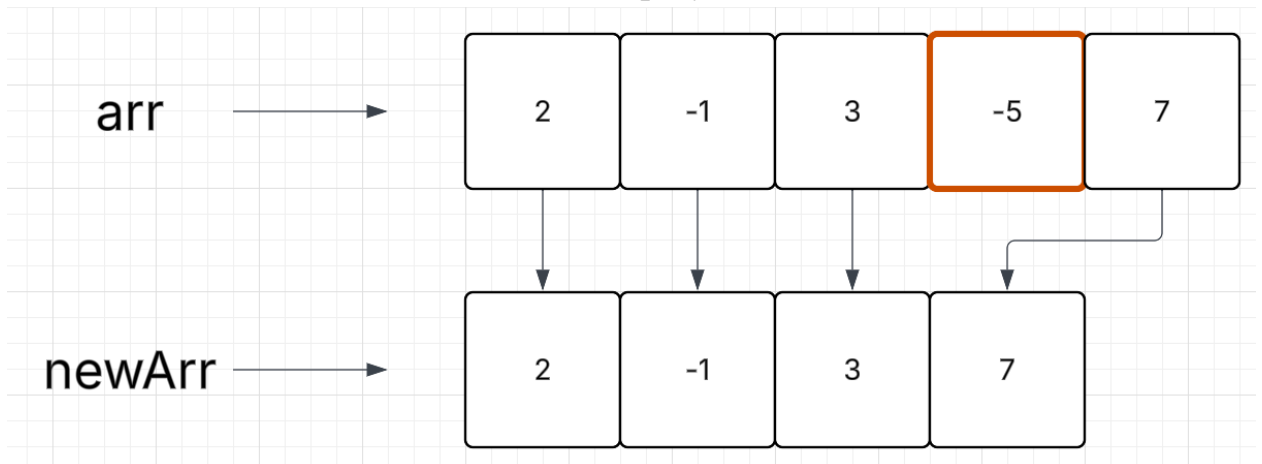


Від'ємний елемент знайшли, але перевіряємо далі



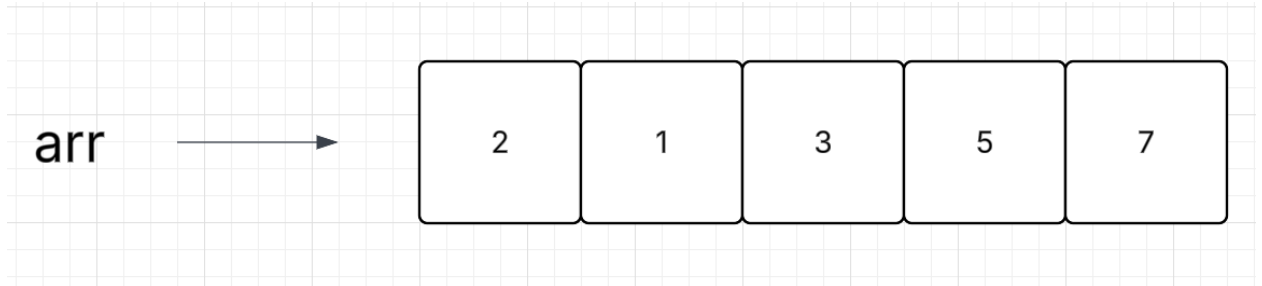
Останній від'ємний елемент знайдено

Виводимо результат

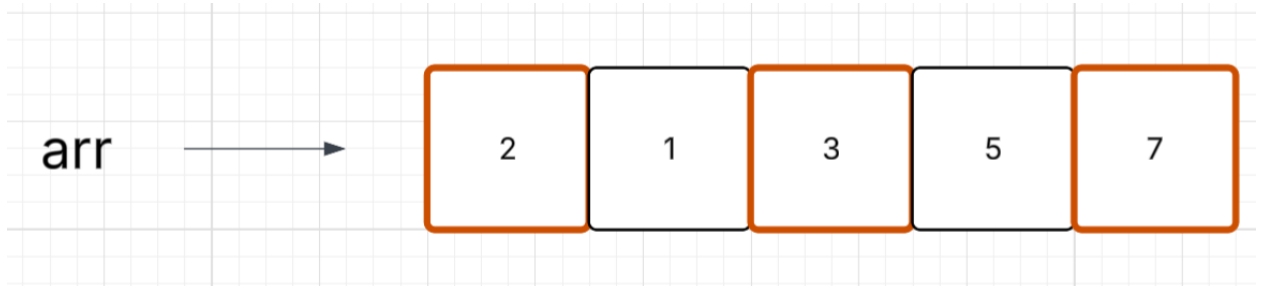


## IgnatenkoSolution.Block\_1\_Task\_6

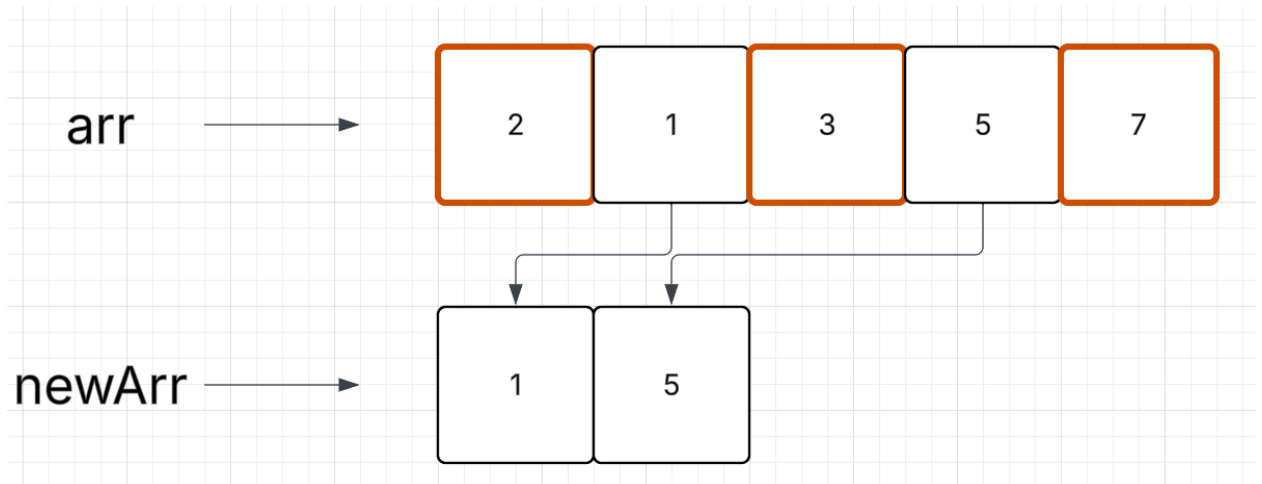
Створюємо одновимірний масив



Видаляємо елементи з парними індексами



Створюємо новий масив і записуємо туди числа з непарними індексами і виводимо результат

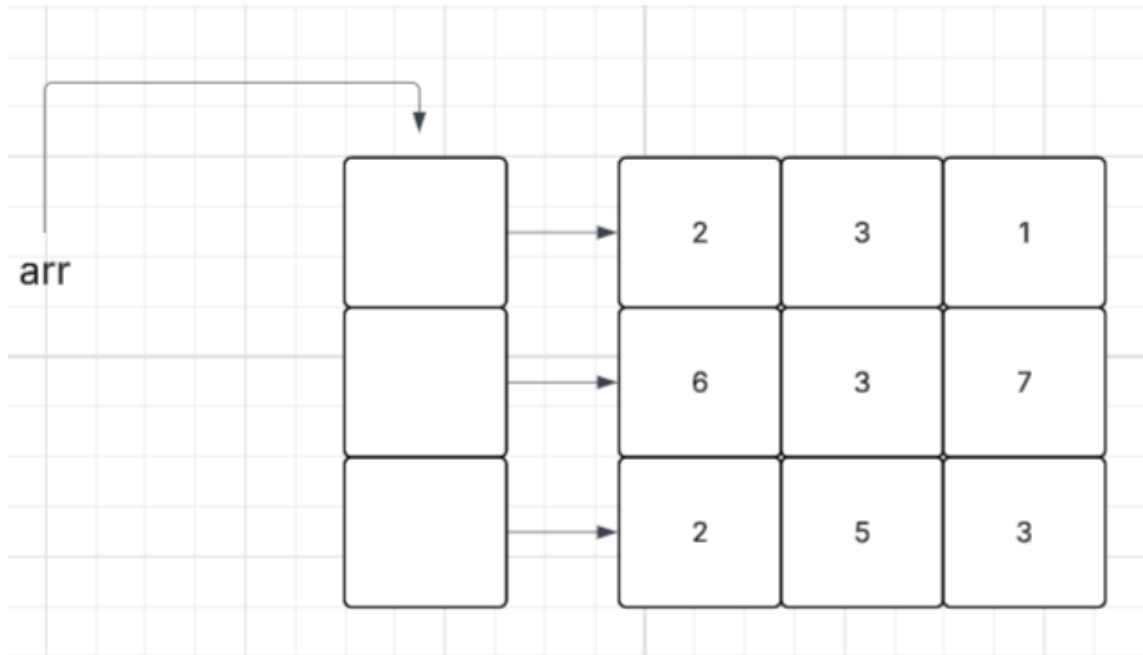




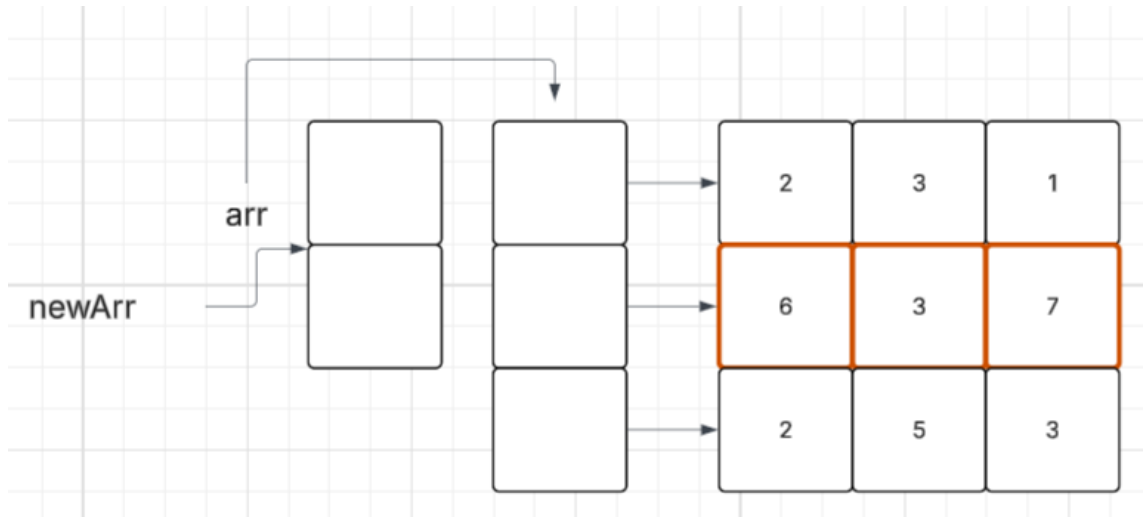
## Блок 2

### PalionniySolution.Block\_2\_Task\_8

Створюємо зубчастий масив

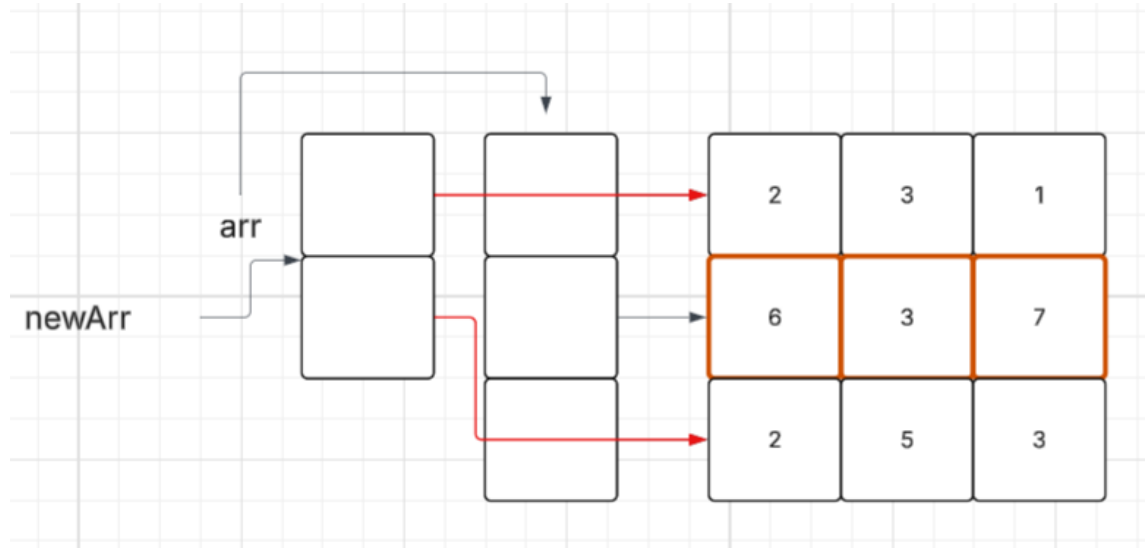


Пошук найбільшого елемента в рядку і створюємо новий зубчастий масив

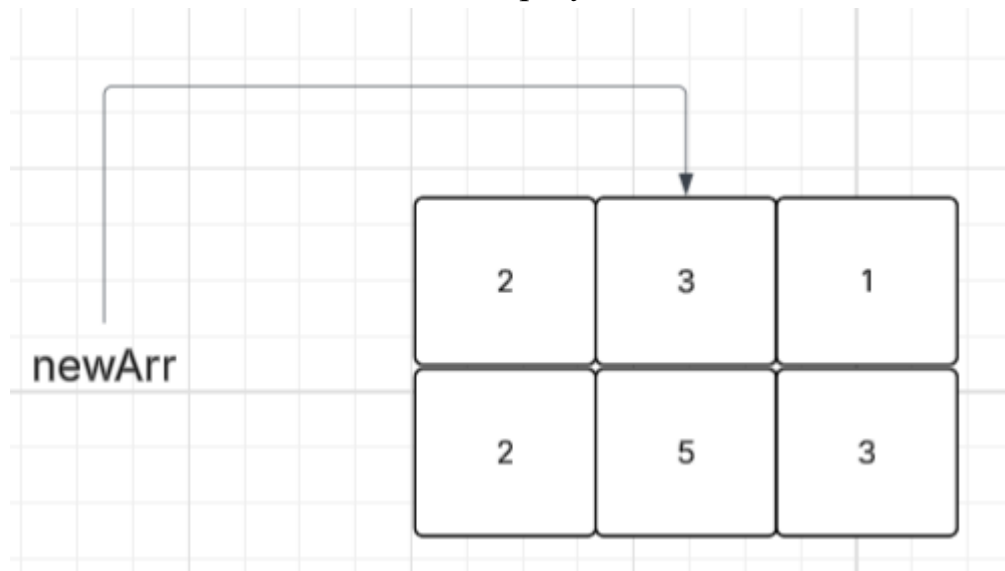


max = 7; index = 2

Записуємо в новий зубчастий масив все окрім рядка з найбільшим індексом

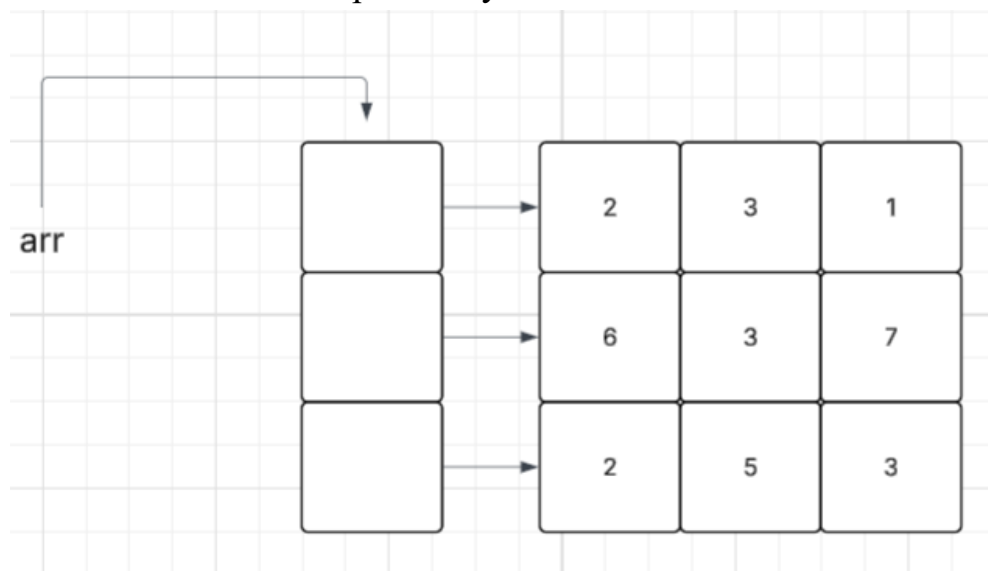


Виводимо результат

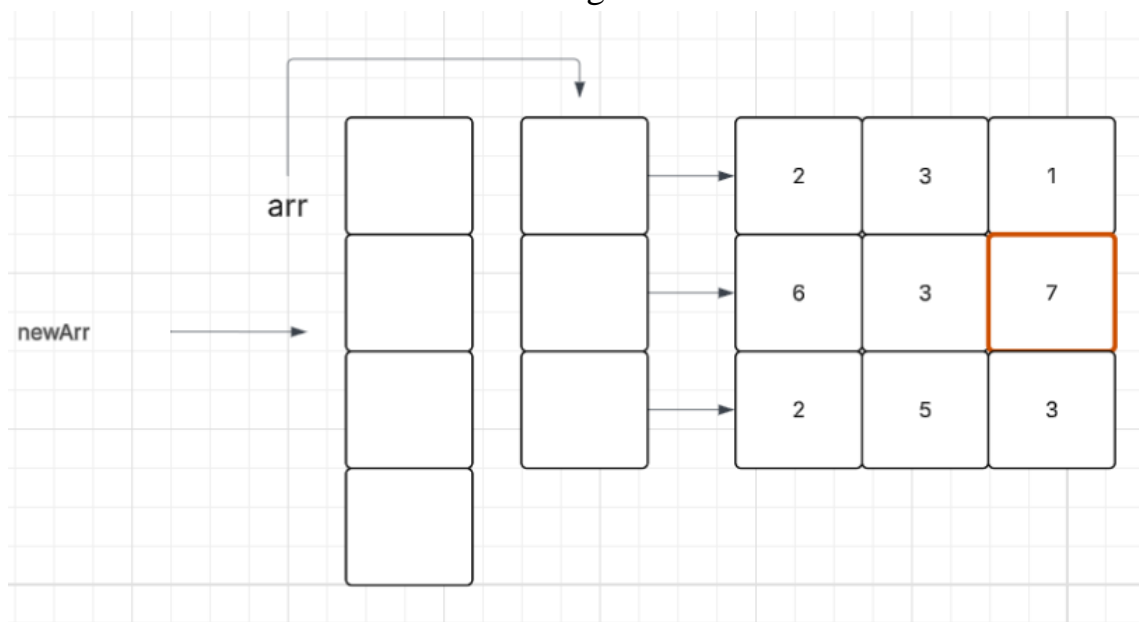


## DinovSolution.Block\_2\_Task\_11

Створюємо зубчастий масив

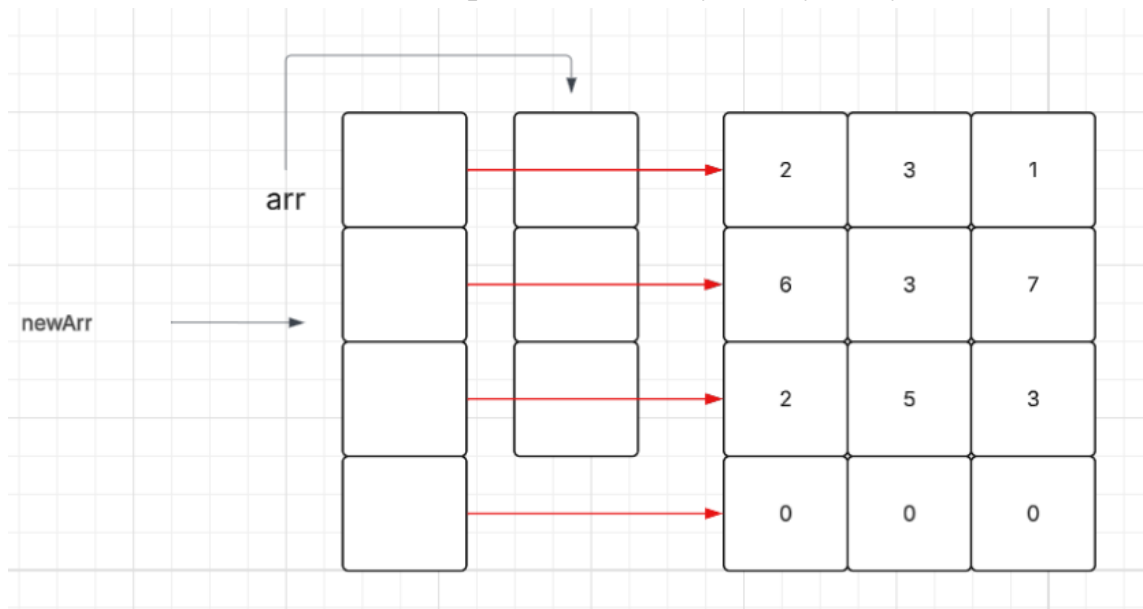


Пошук максимального елемента та створення нового зубчастого масиву з  $arr.Length+1$

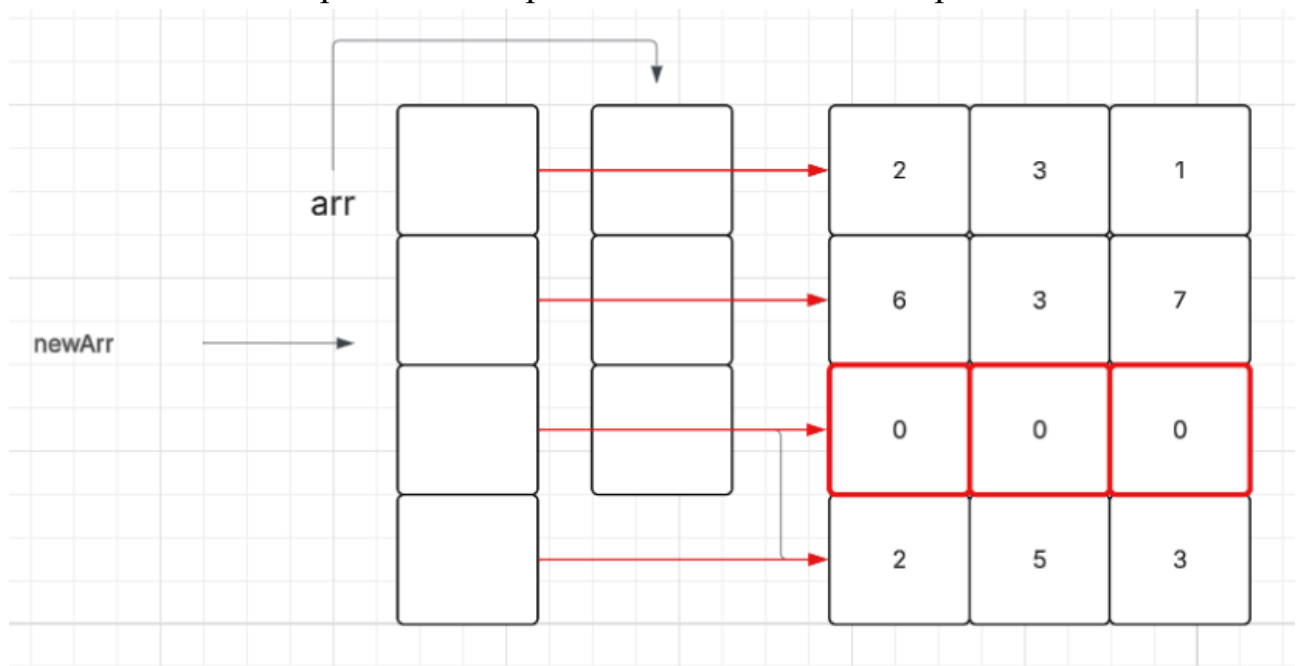


$max = 7; index = 2$

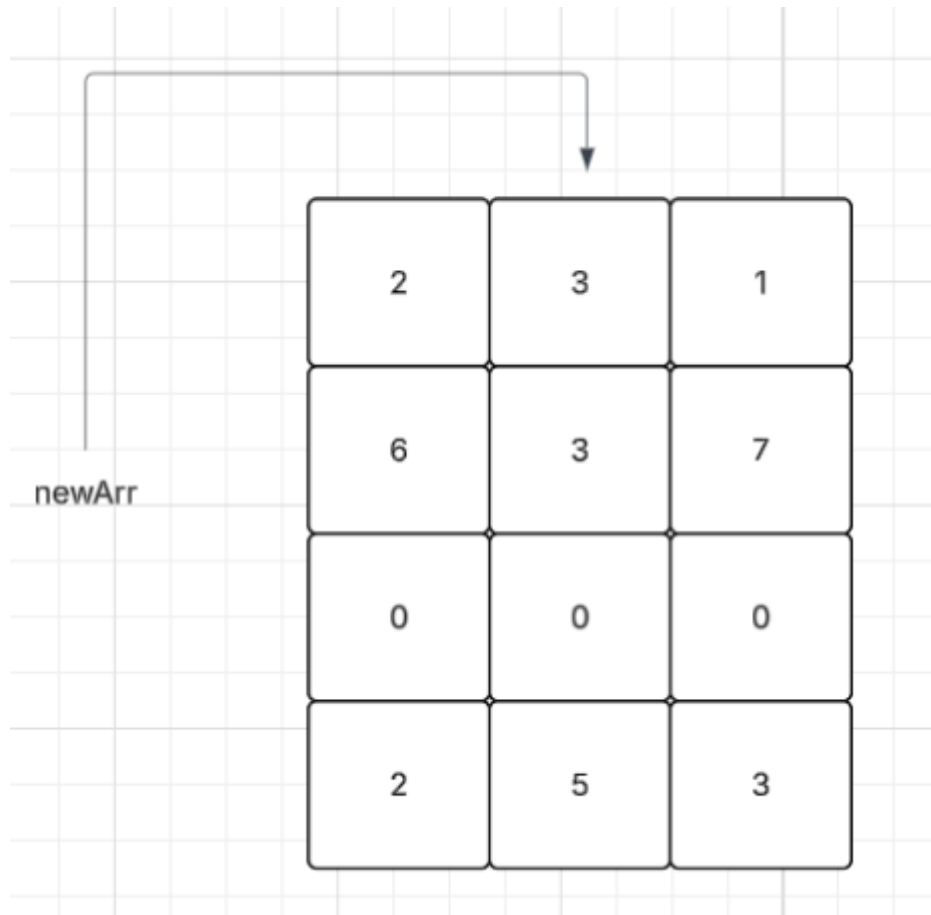
Додаємо новий рядок та записуємо туди нулі



Переставляємо рядок з index+1 на новий рядок

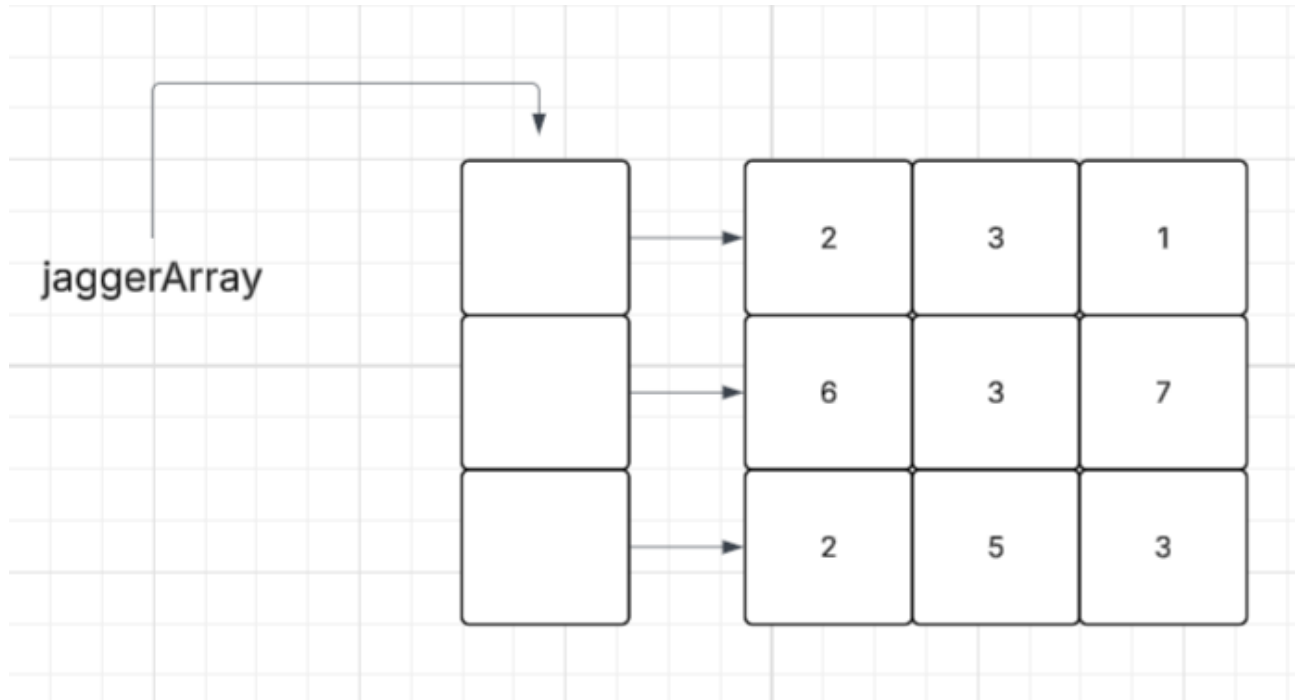


Виводимо результат



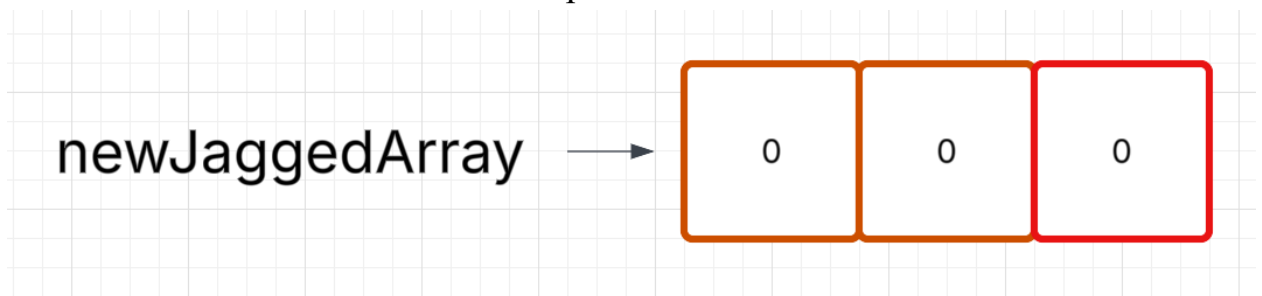
## IgnatenkoSolution.Block\_2\_Task\_2

Створюємо зубчастий масив

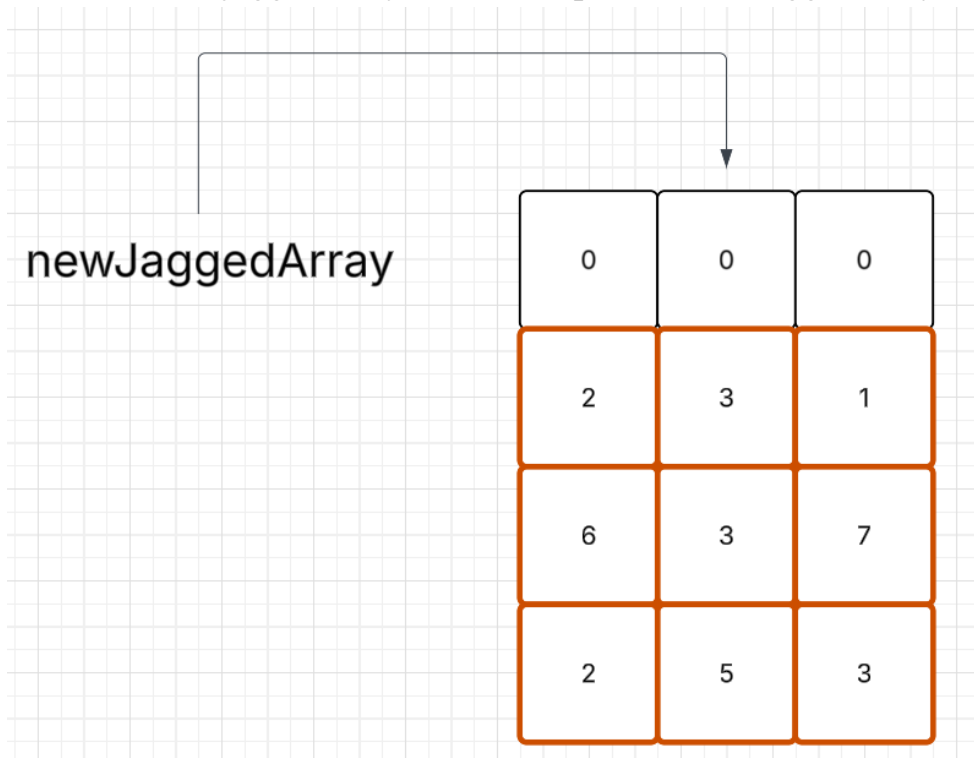


k=1

Створюємо масив



Копіюємо jaggerArray після створеного newJaggerArray



Виводимо результат

