

Завдання 1. Дана квадратна матриця.

Розмістити елементи вказаних стовпців у порядку спадання.

Лістинг програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#define n 5

void sortDescending(int arr[]) {
    int temp;
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (arr[i] < arr[j]) {
                temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int matrix[n][n];

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            matrix[i][j] = rand() % 10;
        }
    }

    printf("Початкова матриця:\n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }

    int columns_to_sort[] = { 0, 1, 2, 3, 4 };

    for (int k = 0; k < sizeof(columns_to_sort) / sizeof(columns_to_sort[0]); k++) {
        int col_index = columns_to_sort[k];
        int column_values[n];

        for (int i = 0; i < n; i++) {
            column_values[i] = matrix[i][col_index];
        }
    }
}
```

					ДУ «Житомирська політехніка» 23.121.15.000 – Лр 15			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Нагорний Т. Г.			Звіт з лабораторної роботи		Лім.	Арк.
Перевір.		Прохорчук Д. В.						1
Керівник								9
Н. контр.							ФІКТ Гр. ВТ-23-1[1]	
Зав. каф.								

```

        sortDescending(column_values, n);

        for (int i = 0; i < n; i++) {
            matrix[i][col_index] = column_values[i];
        }
    }

    printf("\nМатриця після сортування вказаних стовпців у порядку спадання:\n");
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }

    return 0;
}

```

```

Початкова матриця:
1 7 4 0 9
4 8 8 2 4
5 5 1 7 1
1 5 2 7 6
1 4 2 3 2

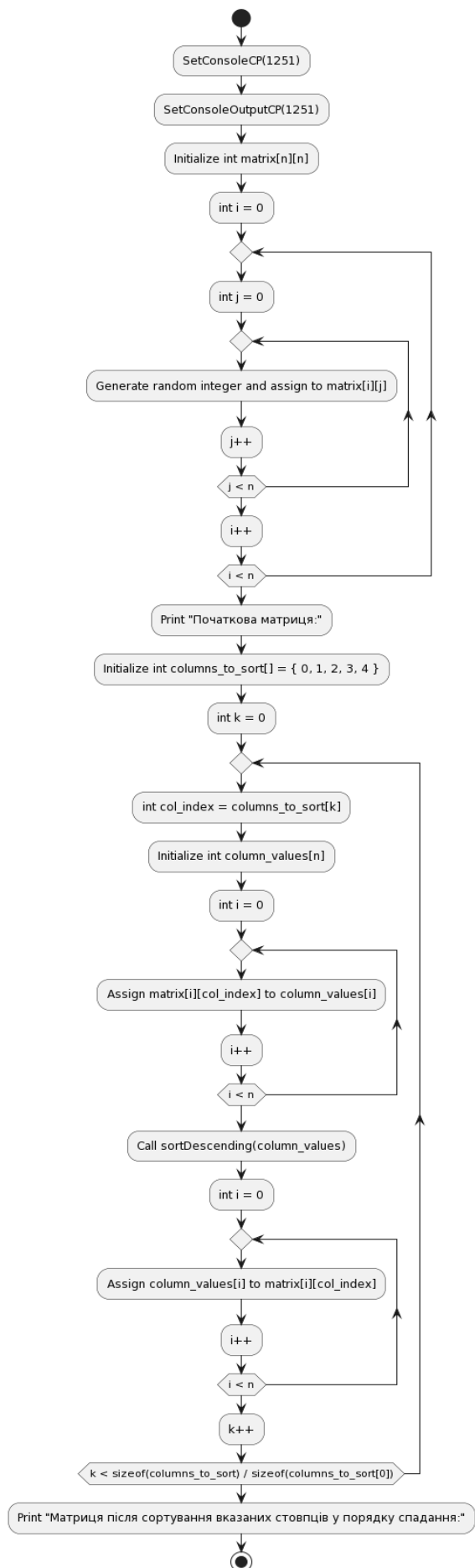
Матриця після сортування вказаних стовпців у порядку спадання:
5 8 8 7 9
4 7 4 7 6
1 5 2 3 4
1 5 2 2 2
1 4 1 0 1

```

Результат виконання програми

		Нагорний Т. Г.			ДУ «Житомирська політехніка».23.121.15.000 – Лр15	Арк.
		Прохорчук Д. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

Блок схема:



Завдання 2. Дана прямокутна матриця.

Ущільнити задану матрицю, вилучаючи із неї рядок і стовпець де знаходиться її мінімальний елемент.

Лістинг програми:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <windows.h>

void findMinElementIndex(int matrix[][100], int rows, int cols, int* minRowIndex, int* minColIndex) {
    int minElement = matrix[0][0];
    *minRowIndex = 0;
    *minColIndex = 0;

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (matrix[i][j] < minElement) {
                minElement = matrix[i][j];
                *minRowIndex = i;
                *minColIndex = j;
            }
        }
    }
}

void compressMatrix(int matrix[][100], int rows, int cols, int minRowIndex, int minColIndex) {
    for (int i = minRowIndex; i < rows - 1; i++) {
        for (int j = 0; j < cols; j++) {
            matrix[i][j] = matrix[i + 1][j];
        }
    }
    rows--;

    for (int j = minColIndex; j < cols - 1; j++) {
        for (int i = 0; i < rows; i++) {
            matrix[i][j] = matrix[i][j + 1];
        }
    }
    cols--;

    printf("\nУщільнена матриця після вилучення рядка та стовпця з мінімальним елементом:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }
}

int main() {
    int rows, cols;
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    printf("Введіть кількість рядків та стовпців (розділені пробілом): ");
    scanf_s("%d %d", &rows, &cols);
```

		Нагорний Т. Г.			ДУ «Житомирська політехніка».23.121.15.000 – Лр15	Арк.
		Прохорчук Д. В.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if (rows > 100 || cols > 100) {
    printf("Перевищено максимальний розмір матриці.\n");
    return 1;
}

int matrix[100][100];

srand(time(NULL));
printf("Початкова матриця:\n");
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        matrix[i][j] = rand() % 9 + 1;
        printf("%d ", matrix[i][j]);
    }
    printf("\n");
}

int minRowIndex, minColIndex;
findMinElementIndex(matrix, rows, cols, &minRowIndex, &minColIndex);
compressMatrix(matrix, rows, cols, minRowIndex, minColIndex);

return 0;
}

```

```

Введіть кількість рядків та стовпців (розділені пробілом): 2 4
Початкова матриця:
7 2 1 6
9 4 9 9

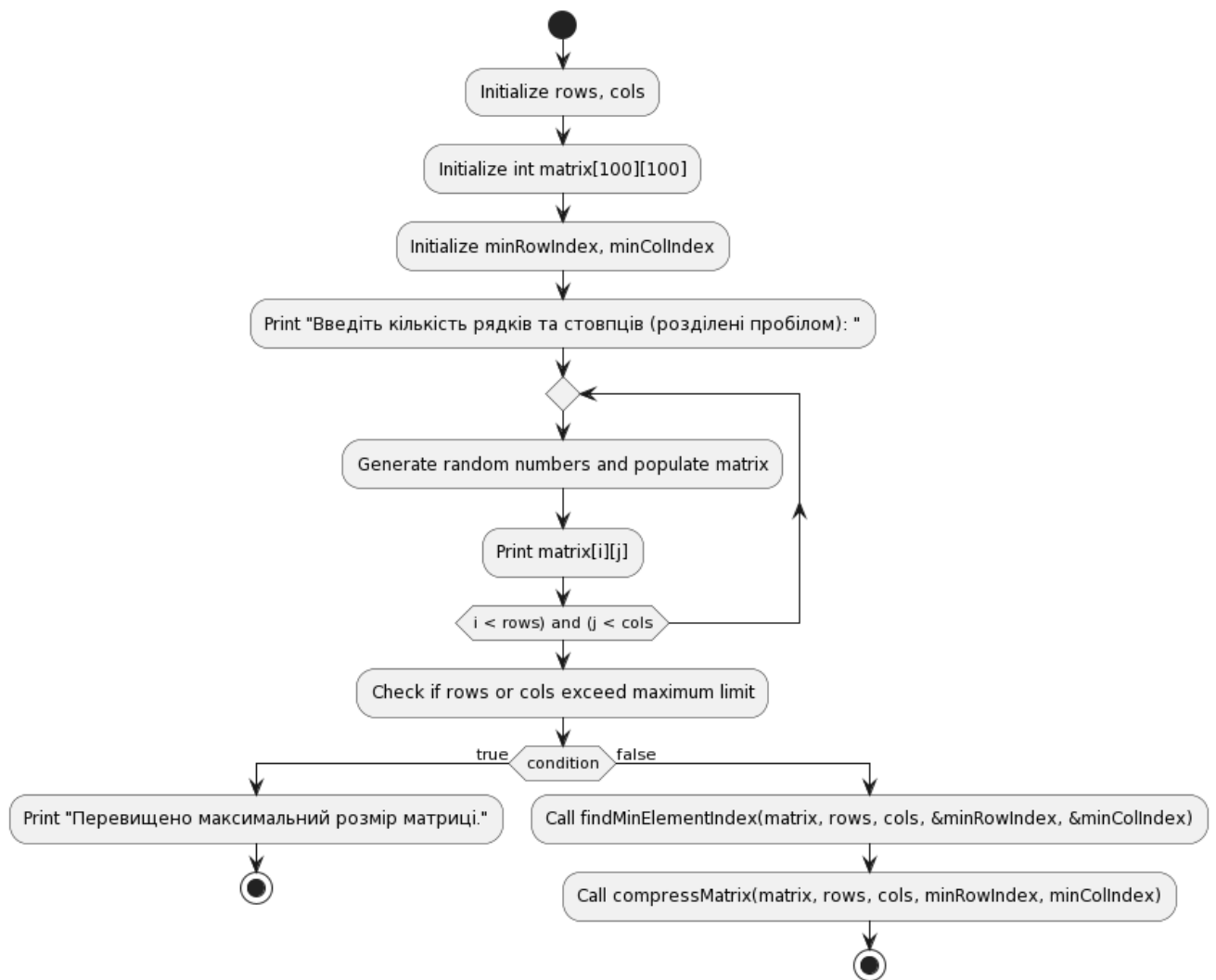
Ущільнена матриця після вилучення рядка та стовпця з мінімальним елементом:
9 4 9

```

Результат виконання програми

		Нагорний Т. Г.			ДУ «Житомирська політехніка».23.121.15.000 – Лр15	Арк.
		Прохорчук Д. В.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

Блок схема:



Завдання 3. Здійснити сортування двовимірного масиву з наступними параметрами (за допомогою додаткового одновимірного масиву):

обміном,
вставками

ціле

За спаданням

Лістинг програми:

```
#include <stdio.h>
#include <windows.h>

void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j] < arr[j + 1]) {
                int temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

void insertionSort(int arr[], int n) {
    int i, key, j;
    for (i = 1; i < n; i++) {
        key = arr[i];
        j = i - 1;

        while (j >= 0 && arr[j] < key) {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}

void printArray(int arr[], int n) {
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");
}

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    int rows, cols;

    printf("Введіть кількість рядків та стовпців (розділені пробілом): ");
    scanf_s("%d %d", &rows, &cols);

    int matrix[100][100];

    printf("Введіть елементи матриці:\n");
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            scanf_s("%d", &matrix[i][j]);
        }
    }

    printf("Початкова матриця:\n");
```

```

for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        printf("%d ", matrix[i][j]);
    }
    printf("\n");
}

int tempArray[100 * 100];
int index = 0;

for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        tempArray[index++] = matrix[i][j];
    }
}

bubbleSort(tempArray, rows * cols);

printf("\nВідсортований методом обміну за спаданням:\n");
printArray(tempArray, rows * cols);

insertionSort(tempArray, rows * cols);

printf("\nВідсортований методом вставок за спаданням:\n");
printArray(tempArray, rows * cols);

return 0;
}

```

```

Введіть кількість рядків та стовпців (розділені пробілом): 3 5
Введіть елементи матриці:
1 2 3
1 3 4
5 4 3
2 34 4
3 2 12
Початкова матриця:
1 2 3 1 3
4 5 4 3 2
34 4 3 2 12

Відсортований методом обміну за спаданням:
34 12 5 4 4 4 3 3 3 3 2 2 2 1 1

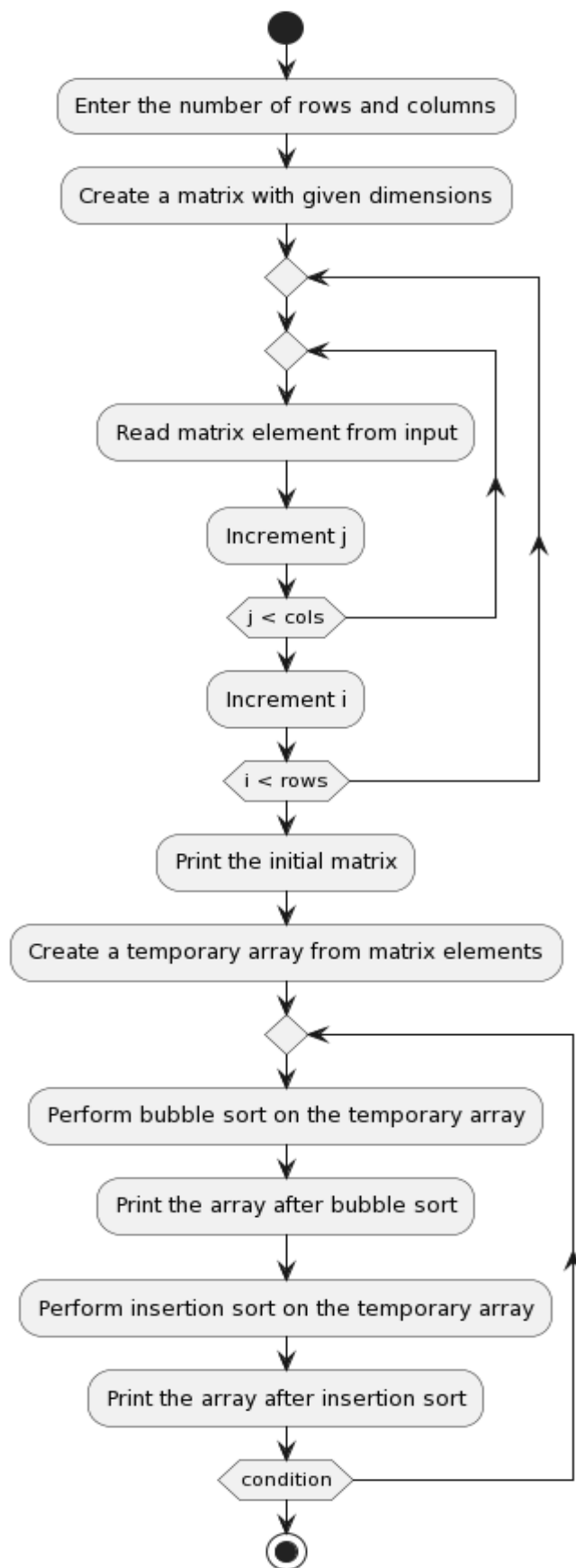
Відсортований методом вставок за спаданням:
34 12 5 4 4 4 3 3 3 3 2 2 2 1 1

```

Результат виконання програми

		Нагорний Т. Г.			ДУ «Житомирська політехніка».23.121.15.000 – Лр15	Арк.
		Прохорчук Д. В.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

Блок схема:



Висновок: набуто практичних навичок роботи з двовимірними масивами.