

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

ІКТА



З В І Т

**до лабораторних робіт з курсу: ПРОГРАМУВАННЯ, ЧАСТИНА 2
(ОБ'ЄКТНООРІЄНТОВАНЕ ПРОГРАМУВАННЯ)**

ЛАБОРАТОРНА РОБОТА №3

**ЗАСОБИ РОБОТИ З ДИНАМІЧНОЮ ПАМ'ЯТТЮ. ДИНАМІЧНІ
МАСИВИ**

Оцінка:

Виконав:

Студент групи КІ-103

Кобзєв Р.Д

Роботу прийняв:

Аспірант Науличний В.В

Львів – 2024

Мета роботи:

Познайомитися із динамічними масивами.

Індивідуальне завдання:

Завдання №8: Задано квадратну матрицю, всі елементи якої рівні одиниці. Написати функцію `void func (int** arr, int n)`, котра заповняє заштриховану область матриці (згідно варіанту) нулями. Розмір масиву вводить з клавіатури.



Пояснення:

Створимо динамічну квадратну матрицю з розмірністю, яку введе юзер. Для початку заповнимо її «0», а далі будемо на місці «малюнку» ставити одиниці. Логіка заповнення буде такою: у нас буде змінна типу `int` з назвою `end`. Існує два варіанти:

1. Матриця розмірністю парною
2. Матриця непарною розмірністю

Рядки будуть рухатись від 0 до `n-1`, проте стовпці матимуть алгоритм заповнення. Кожен стовпець від `i-end` буде заповнюватись одиницею. З кожним колом `end` буде збільшуватись на один, допоки не дійде до середини матриці. Таким чином перший рядок буде заповнений на 1 клітку одиницями, другий на 2 з кінця, третій на 3 з кінця і так до середини. Від середини `end` буде зменшуватись, отже `j` буде рости

Код

```
#include <iostream>
```

```
using namespace std;
```

```
void func(int** arr, int number)
{
    int point = number / 2;
    if (number % 2 == 0) {
        int end = number - 1, if_second = 1;
        for (int i = 0; i < number; i++) {
            int old_end = end;
            for (int j = 0; j < number; j++) {
                if (j != end)
                    arr[i][j] = 1;
                else {
                    arr[i][j] = 0;
```

```

        end++;
    }
}
end = old_end;
if (end == point && if_second != 2) {
    if_second++;
} else if (if_second == 2) {
    end++;
} else {
    end--;
}
}
} else {
    int end = number - 1;
    bool if_second = false;
    for (int i = 0; i < number; i++) {
        int old_end = end;
        for (int j = 0; j < number; j++) {
            if (j != end)
                arr[i][j] = 1;
            else {
                arr[i][j] = 0;
                end++;
            }
        }
        end = old_end;
        if (end == point || if_second == true) {
            end++;
            if_second = true;
        } else {
            end--;
        }
    }
}
}

```

```

for (int i = 0; i < number; i++) {
    for (int j = 0; j < number; j++) {
        cout << arr[i][j];
    }
}

```

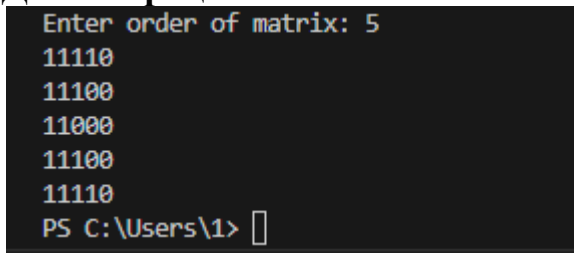
```

        cout << endl;
    }
}

int main()
{
    int number;
    cout<<"Enter order of matrix: ";
    cin>>number;
    int** arr = new int* [number];
    for(int i=0; i<number; i++){
        arr[i] = new int [number];
    }
    func(arr, number);
}

```

Демонстрація:



```

Enter order of matrix: 5
11110
11100
11000
11100
11110
PS C:\Users\1>

```

Висновок:

Виконання даного завдання дозволило закріпити навички роботи з динамічними масивами у мові програмування C++, а також реалізувати алгоритми для модифікації матриць. Написання функції `void func(int** arr, int n)` стало практичним прикладом застосування знань з обробки двовимірних масивів і маніпулювання їхніми елементами.

Загалом, виконане завдання є корисним у розумінні роботи з динамічною пам'яттю та матрицями, що є важливим аспектом програмування на C++.