

**Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря
Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки**

Лабораторна робота №1.6
з дисципліни
«Алгоритми і структури даних»

Виконала:

Студентка групи ІМ-41
Куц Анна Василівна номер
у списку групи: 11

Перевірила:

Молчанова А. А.

Київ 2024

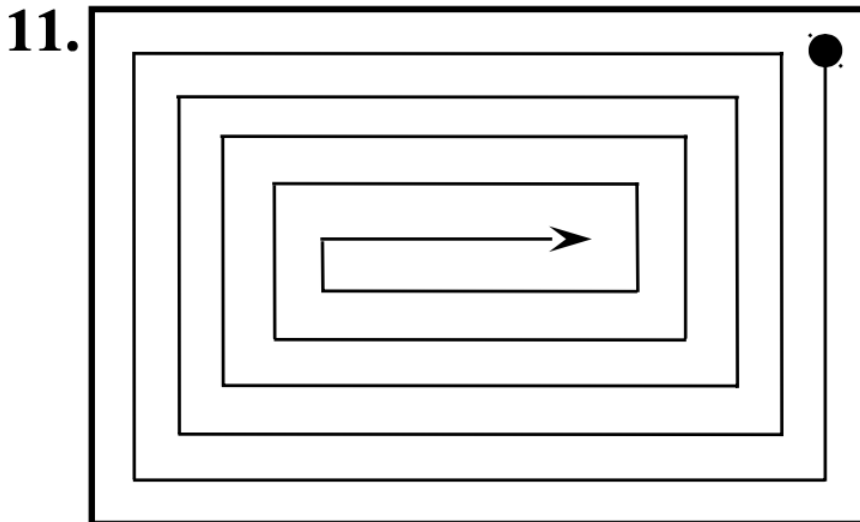
Постановка задачі

1. Оскільки екран монітора має площинні координати так само, як і двовимірний масив (матриця), але, на відміну від останнього, надає можливість візуально спостерігати виконання способу обходу, дана лабораторна робота виконується в координатах екрану монітора (дивись методичні вказівки до виконання даної лабораторної роботи).

2. Завданням даної лабораторної роботи є виконання заданого за варіантом способу обходу на екрані монітору в текстовому режимі, проставляючи довільний символ клавіатури (наприклад ‘*’) у порядку заданого способу обходу.

3. Оскільки при виводі символу у правий ніжній кут екрану відбувається зсув зображення на один рядок вгору (якщо тільки не використовується прямий доступ до відеопам’яті), *останній рядок екрану монітора при виконанні завдання заповнювати не треба.*

Завдання



Текст програми

```
#include <stdio.h>
#include <ncurses.h>
#include <unistd.h>
```

```
int min(int a, int b);  
void draw(int x, int y);
```

```
int main()  
{  
    initscr();  
    noecho();  
    start_color();  
    init_pair(1, COLOR_MAGENTA, COLOR_BLACK);
```

```
    int m, n;  
    getmaxyx(stdscr, m, n);
```

```
    int yn = m - 1;  
    int xn = n - 1;  
    for (int p = 0; p < min(m, n) / 2; p++)  
    {  
        // Moving down  
        for (int i = p; i < yn - p; i++)  
        {  
            draw(i, xn - p);  
        }  
        // Moving left  
        for (int j = xn - p; j > p; j--)  
        {  
            draw(yn - p, j);  
        }  
        // Moving up  
        for (int i = yn - p; i > p; i--)  
        {  
            draw(i, p);  
        }  
        // Moving right  
        for (int j = p; j < xn - p; j++)  
        {  
            draw(p, j);  
        }  
    }
```

```
    if (min(m, n) % 2 != 0)  
    {  
        if (m < n)
```

```

    {
        for (int j = min(m, n) / 2; j <= xn - min(m, n) / 2; j++) {
            draw(m / 2, j);
        }
    } else if (m > n) {
        for (int i = min(m, n) / 2; i <= yn - min(m, n) / 2; i++) {
            draw(i, n / 2);
        }
    }
}

getch();
endwin();
return 0;
}

int min(int a, int b)
{
    return a < b ? a : b;
}

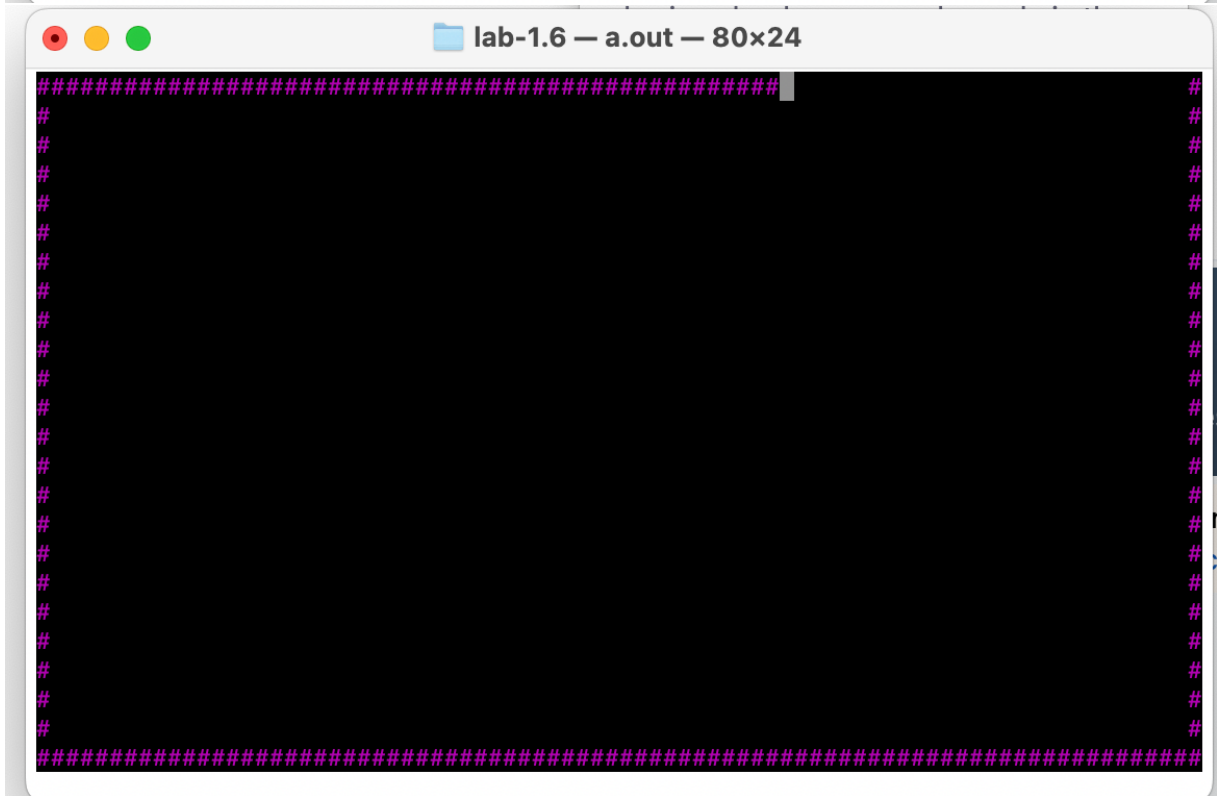
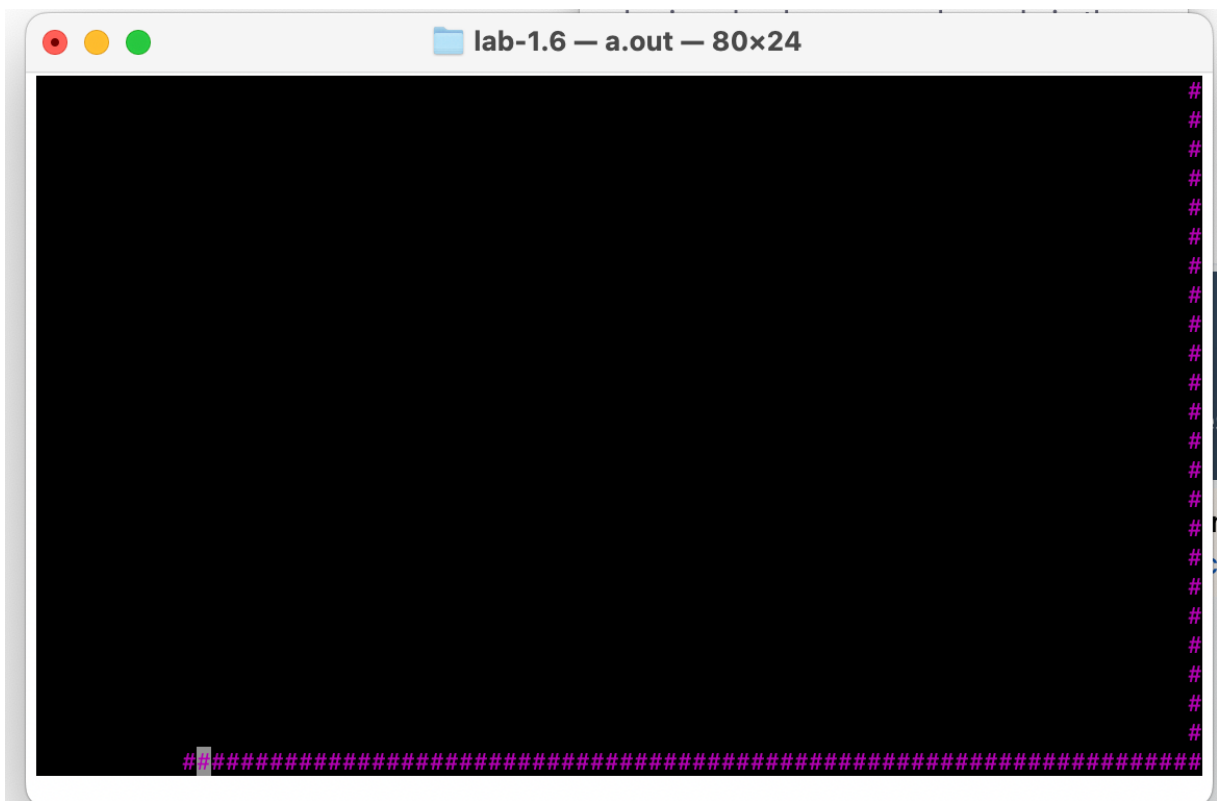
void draw(int x, int y)
{
    attron(COLOR_PAIR(1));
    mvaddch(x, y, '#');
    refresh();
    usleep(20000);
}

```

Тестування програми

Bideo:

https://drive.google.com/file/d/1JQQ1D4XmBhKLwYv_kokEyfT_WaBnKHPm/view?usp=sharing







Висновок

Завдяки виконанню лабораторної роботи №1.6 я засвоїла теоретичний матеріал та набула практичних навичок гнучкої роботи при реалізації алгоритмів з використанням двовимірних масивів. Було створено програму для обходу буферу консолі комп'ютера. Для цього я використала бібліотеку `ncurses` для користувачів UNIX-подібних операційних систем. Були використані вкладені цикли, де коефіцієнт p – номер оберта, контролює вивід символів всередині двовимірного масиву. До прикладу, розмір консолі

80 на 24, отже програма 12 разів повторює заданий алгоритм для обходження буферу по спіралі, адже за допомогою функції `min` шукається найменше значення к-сті рядків чи стовпців та ділиться на 2. Якщо значення `min` є непарним, то в кінці програми ця умова перевіряється та дописується координати, яких не вистачає. Я зробила перевірку, якщо висота менша за ширину, то додається відповідний рядок. Якщо висота більша за ширину, то додається відповідний стовпець. `yn` та `xn` – це максимальні значення рядків та стовпців матриці відповідно. Під час руху донизу(`Moving down`) збільшується значення рядків, а значення стовпців лишаються незмінними. Під час руху наліво(`Moving left`) значення стовпчиків декрементуються, а значення рядку лишається незмінним. Під час руху доверху(`Moving up`) значення стовпців залишається незмінним, а значення рядків декрементуються. Під час руху вправо(`Moving right`) значення рядку є константним, а значення стовпців інкрементується.