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 №4

з дисципліни «Алгоритми і структури даних»

Виконав:

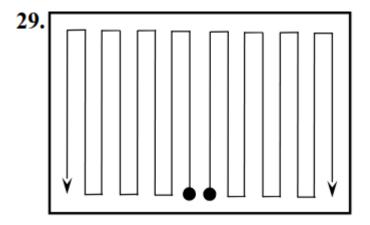
студент групи IM-42 Туров Андрій Володимирович номер у списку групи: 29 Перевірила:

Молчанова А. А.

Завдання

- 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 екрана монітора (дивись методичні вказівки до виконання даної лабораторної роботи).
- 2. Завданням даної лабораторної роботи є виконання заданого за варіантом способу обходу на екрані монітора в текстовому режимі, проставляючи довільний символ клавіатури (наприклад '*') у порядку заданого способу обходу.
- 3. Оскільки при виводі символу у правий ніжній кут екрана відбувається зсув зображення на один рядок вгору (якщо тільки не використовується прямий доступ до відеопам'яті), останній рядок екрана монітора при виконанні завдання заповнювати не треба.

Варіант 29



Текст програми

```
#include <curses.h>
#include <unistd.h>

const int REFRESH_DELAY = 20000;

void get_symbol(int *symbol) {
    printw("Enter the symbol and press Enter:");

    // Ask for the symbol until Enter is pressed
    int next_symbol;
    for (;;) {
        next_symbol = getch();
        if (next_symbol == KEY_ENTER || next_symbol == '\n' || next_symbol == '\r') {
        break;
    } else if (next_symbol == KEY_BACKSPACE) {
        mvprintw(0, 34, " ");
    } else {
```

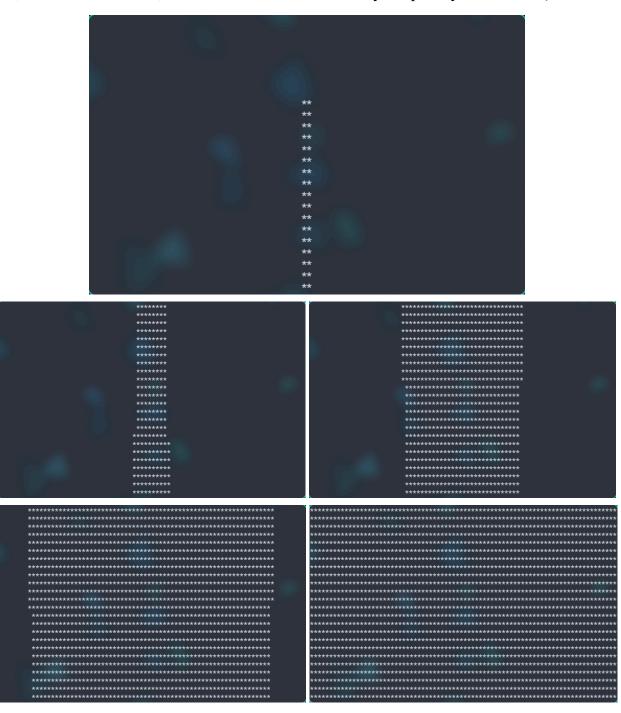
```
mvprintw(0, 34, "%c ", next_symbol);
                    *symbol = next_symbol;
             }
      }
}
void draw_loop(int height, int width, int symbol) {
      const char WIDTH_IS_ODD = width % 2 == 1;
      char direction = -1;
      for (
             int x = 0.5*width - 1 + WIDTH_IS_ODD; // middle
             (x < width) \&\& (x >= 0); // out of bounds check
             X--
      ) {
             for (
                    int y = (direction == 1) ? 0 : height - 1; // whether to start
from top or bottom
                    (y < height) \&\& (y >= 0); // out of bounds check
                    y += direction
             ) {
                    mvaddch(y, x, symbol); // left point
                    refresh();
                    usleep(REFRESH_DELAY);
                    // there was a check not to draw the second point if width is odd
and x == 0,
                    // but mvaddch macro already does that for me
                    mvaddch(y, width - x - 1 + WIDTH_IS_ODD, symbol); // right point,
mirrored
                    refresh();
                    usleep(REFRESH_DELAY);
             }
             direction *= -1;
      }
}
int main(void) {
      initscr();
      noecho();
      keypad(stdscr, TRUE);
      curs_set(FALSE);
      int height, width;
      getmaxyx(stdscr, height, width);
      int symbol;
      get_symbol(&symbol);
      clear();
      refresh();
      draw_loop(height, width, symbol);
```

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

Результати тестування програми

https://asciinema.org/a/gv2iLtedSsyUlsIFk070MBBgM - демонстрація роботи.

(Насправді дві стрілки заповнюються по черзі, але тут недостатньо високий фпс, щоб це помітити, а ставити зовсім аж велику затримку я не хотів.)



Висновки

Використав бібліотеку ncurses, щоб показати обхід екрана як двомірного масиву і його заповнення довільним символом у заданому порядку, зважаючи також на непарність довжини термінала.