Міністерство освіти і науки України

Національний університет «Одеська політехніка»

Навчально-науковий інститут комп’ютерних систем

Кафедра інформаційних систем

Лабораторна робота № 16

**З дисципліни**: «Алгоритмізація та програмування»

Тема: «Управління виведенням даних на екран»

Варіант № 5

**Виконав:**

Студент групи AI-243

Гаврилов О. В.

**Перевірили:**

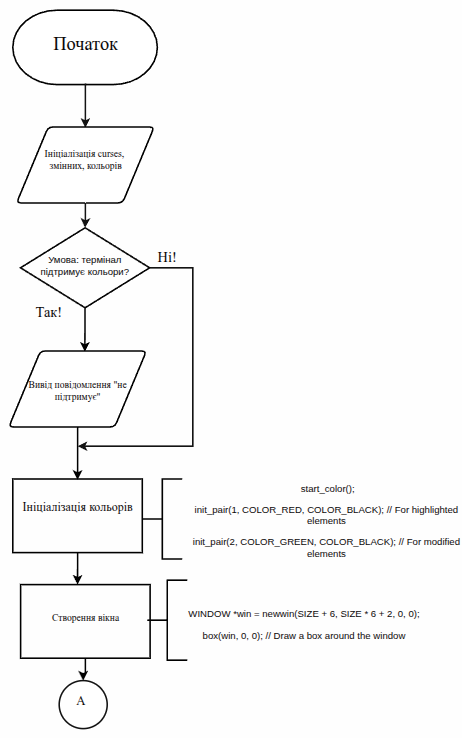
Іванов О.В.

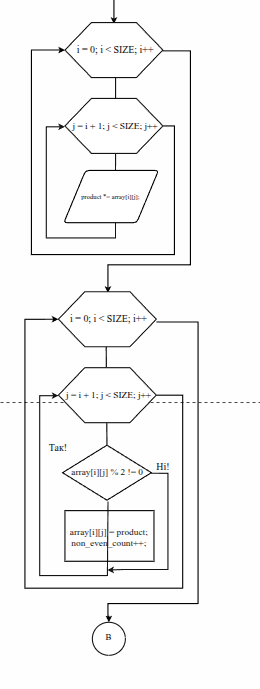
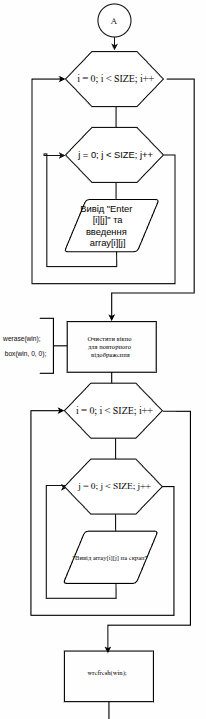
##### Манікаєва О.С.

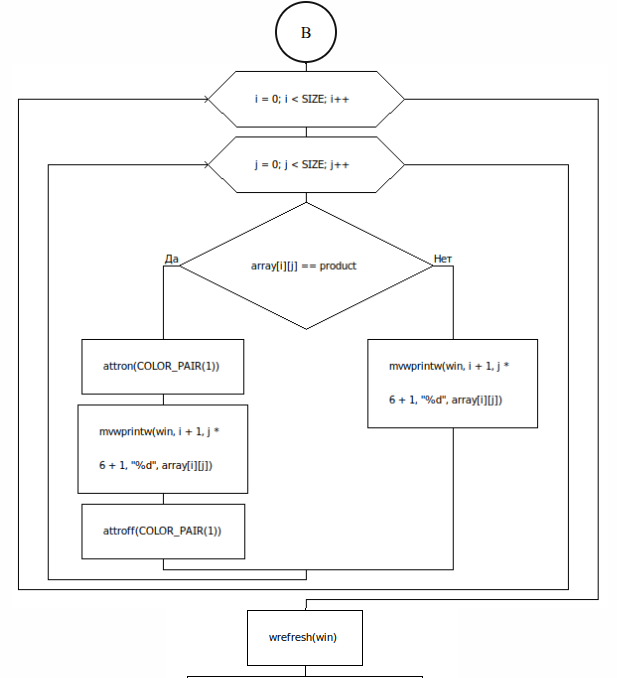
**Мета роботи**: отримання навичок роботи з функціями виводу даних на екран в текстовому режимі.

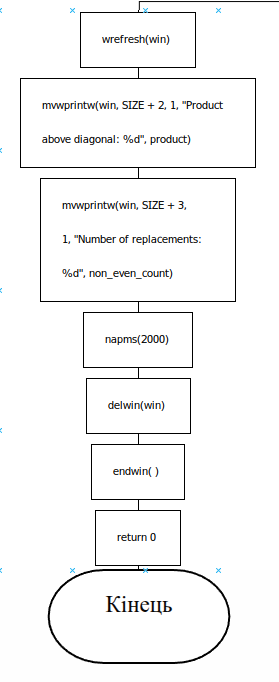
***Завдання 16.1***

***Блок-схема алгоритму наведена на рисунку 16.1***

******



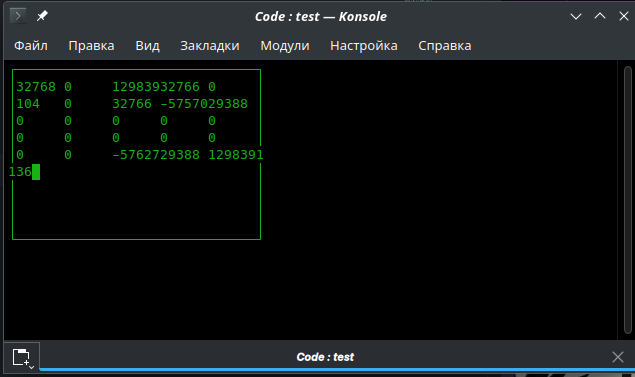


Рисунок 16.1 – Блок-схеми алгоритму виконання завдання

**Програмний код:**

|  |
| --- |
| #include <curses.h>  #include <stdlib.h>  #define SIZE 5  **int** main() {  **int** array[SIZE][SIZE];  **int** i, j;  **int** product = 1;  **int** non\_even\_count = 0;  // Initialize curses  initscr();  cbreak();  noecho();  keypad(stdscr, TRUE);  // Check if the terminal supports color  **if**(has\_colors() == FALSE) {  endwin();  printf("Your terminal does not support color\n");  exit(1);  }  start\_color();  init\_pair(1, COLOR\_RED, COLOR\_BLACK); // For highlighted elements  init\_pair(2, COLOR\_GREEN, COLOR\_BLACK); // For modified elements  // Create a window for input and display  WINDOW \*win = newwin(SIZE + 6, SIZE \* 6 + 2, 0, 0);  box(win, 0, 0); // Draw a box around the window  // Input elements  **for**(i = 0; i < SIZE; i++) {  **for**(j = 0; j < SIZE; j++) {  mvwprintw(win, i + 1, j \* 6 + 1, "Enter [%d][%d]: ", i, j);  wrefresh(win);  scanw("%d", &array[i][j]);  }  }  // Clear the window for redisplay  werase(win);  box(win, 0, 0);  // Display the array  **for**(i = 0; i < SIZE; i++) {  **for**(j = 0; j < SIZE; j++) {  mvwprintw(win, i + 1, j \* 6 + 1, "%d", array[i][j]);  }  }  wrefresh(win);  // Calculate the product of elements above the main diagonal  **for**(i = 0; i < SIZE; i++) {  **for**(j = i + 1; j < SIZE; j++) {  product \*= array[i][j];  }  }  // Replace non-even elements with the product  **for**(i = 0; i < SIZE; i++) {  **for**(j = 0; j < SIZE; j++) {  **if**(array[i][j] % 2 != 0) {  array[i][j] = product;  non\_even\_count++;  }  }  }  // Pause for 2 seconds  napms(2000);  // Change the color of the window to green  wbkgd(win, COLOR\_PAIR(2));  werase(win);  box(win, 0, 0);  // Display the modified array with modified elements highlighted in red  **for**(i = 0; i < SIZE; i++) {  **for**(j = 0; j < SIZE; j++) {  **if**(array[i][j] == product) {  attron(COLOR\_PAIR(1));  mvwprintw(win, i + 1, j \* 6 + 1, "%d", array[i][j]);  attroff(COLOR\_PAIR(1));  } **else** {  mvwprintw(win, i + 1, j \* 6 + 1, "%d", array[i][j]);  }  }  }  wrefresh(win);  // Print the product and number of replacements  mvwprintw(win, SIZE + 2, 1, "Product above diagonal: %d", product);  mvwprintw(win, SIZE + 3, 1, "Number of replacements: %d", non\_even\_count);  // Wait for another 2 seconds  napms(2000);  // End curses  delwin(win);  endwin();  **return** 0;  } |

**Результати тестування:**



**Висновок:**

У цій програмі ми використовували бібліотеку curses для створення інтерактивного текстового інтерфейсу в консолі Linux. Ми організували введення елементів двовимірного масиву 5х5, обчислили добуток елементів, розташованих вище головної діагоналі, і замінили всі непарні елементи на цей добуток. Після цього ми змінили колір вікна, в якому відображається масив, щоб виділити зміни. Це демонструє, як можна ефективно використовувати curses для створення динамічних і візуально привабливих консольних додатків, які реагують на дії користувача і змінюють свій вигляд відповідно до логіки програми.

**Контрольні питання:**

### **1. Які текстові режими підтримують відеоадаптери VGA/SVGA. Як змінити текстовий режим на інший?**

Відеоадаптери VGA та SVGA підтримують кілька текстових режимів, найпоширенішим з яких є режим 80×25 символів, де кожен символ займає 2 байти (один для коду символу, інший для кольору). Також існують режими 40×25, 80×43, 80×50 тощо. Щоб змінити текстовий режим, використовуються функції BIOS або спеціальні програми, які надсилають відповідний запит відеоадаптеру.

**2. Що називається текстовим вікном?**

Текстовим вікном називається обмежена частина екрана, в якій виводиться текст, незалежно від розміру усього екранного простору. Це може бути як окремий прямокутник у межах екрана, так і повністю весь екран.

### 3. Як відкрити текстове вікно?

Щоб відкрити текстове вікно, потрібно задати його координати — позицію верхнього лівого і нижнього правого кутів у межах екрана, а також параметри кольору тла і символів. Це дозволяє програмно виділити певну область для роботи з текстом.

### 4. За допомогою яких функцій можна керувати позицією і формою текстового курсору?

Керування позицією і формою текстового курсору виконується за допомогою функцій BIOS або низькорівневих засобів, що дозволяють перемістити курсор у задану позицію на екрані або змінити його вигляд, наприклад, зробити його невидимим чи змінити висоту.

**5. На які групи поділяються клавіші клавіатури ПК?**

Клавіші клавіатури поділяються на алфавітно-цифрові, керуючі, функціональні, клавіші керування курсором, клавіші редагування, а також спеціальні клавіші, такі як Shift, Ctrl, Alt, які змінюють функції інших клавіш.