*Додаток 1*

Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 8 з дисципліни

«Основи програмування 1. Базові конструкції»

Варіант №2

Виконав студент Бабіч Денис Володимирович

(шифр, прізвище, ім'я, по батькові)

Перевірив \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище, ім'я, по батькові)

**Лабораторна робота № 8**

**Мета**: опанувати технологію використання двовимірних масивів.

**Завдання**:

**Варіант 2**



**Розв’язок**

1. ***Постановка задачі***

Для розв’язання поставленого завдання застосовуватиметься динамічний двовимірний масив для зберігання результатів роботи функцій. Функція create\_matrix() – за допомогою 2 арифметичних циклів створює та задає новостворену матрицю випадковими числами. Функція *print\_matrix() виводить матрицю на консоль. Функція examine\_matrix() займається тим, що викликає функцію current\_max(), передаючи аргументом саму матрицю, її розмір, там індекс елемента, який буде заповнено (при першому проході циклів викликаю функцію з такими аргументами: current\_max(0, 0, matrix, size)). Сама функція current\_max() порівнює елементи у обраному діапазоні и встановлює який є найбільшим. Поетапний прохід по матриці 3 x 3 виглядає так:*

. . .

. . . . . . .

. . . . . . . . .

. . .

. . . . . . .

. . .

І на кожному встановлюється елемент, який є найбільшим. Саме він і вноситься до нової матриці.

*2.* ***Побудова математичної моделі***

|  |  |  |  |
| --- | --- | --- | --- |
| *Тип данних* | *Змінна* | *Тип* | *Призначення* |
| *int\*\** | *matrix* | *Вказівник на вказівник (ціле число)* | *Двомірний масив* |
| *Змінні у підпрограмах* | | | |
| *create\_matrix()* | | | |
| *Тип данних* | *Змінна* | *Тип* | *Призначення* |
| *int\*\** | *matrix* | *Вказівник на вказівник (ціле число)* | *Двомірний масив* |
| *int* | *size* | *Ціле число* | *Кількість елементів рядках і в стовпчиках матриці* |
| *int* | *lower\_bound* | *Ціле число* | *Нижня межа випадково зегенерованих чисел* |
| *int* | *upper\_bound* | *Ціле число* | *Верхня межа випадково зегенерованих чисел* |
| *print\_matrix()* | | | |
| *int\*\** | *matrix* | *Вказівник на вказівник (ціле число)* | *Двомірний масив* |
| *int* | *size* | *Ціле число* | *Кількість елементів рядках і в стовпчиках матриці* |
| *examine\_matrix()* | | | |
| *int\*\** | *matrix* | *Вказівник на вказівник (ціле число)* | *Двомірний масив* |
| *int* | *size* | *Ціле число* | *Кількість елементів рядках і в стовпчиках матриці* |
| *current\_max()* | | | |
| *int\*\** | *matrix* | *Вказівник на вказівник (ціле число)* | *Двомірний масив* |
| *int* | *row* | *Ціле число* | *Розглядаємий рядок, у якому знаходиться елемент* |
| *int* | *column* | *Ціле число* | *Розглядаємий стовпичк, у якому знаходиться елемент* |
| *int* | *size* | *Ціле число* | *Кількість елементів рядках і в стовпчиках матриці* |
| *int* | *max* | *Ціле число* | *Тимчасова змінна, які присвоюється значення matrix[i][j]* |
| *int* | *first* | *Ціле число* | *Розглядаємий елемент* |
| *int* | *last* | *Ціле число* | *Розглядаємий елемент* |

1. ***Код на С++***









1. ***Перевірка***



*5.* ***Випробовування програми***

|  |  |
| --- | --- |
| Блок | Дія |
|  | **початок** |
|  | Введення: 3  //matrix =  {-88 49 98}  {-32 -35 -10}  { 91 -41 28}  //Виклик функції process\_matrix(0, 0, matrix, 3)  //first\_j\_in\_row = 0, last\_j\_in\_row = 0,  //max = matrix[0][0], max = -88  //повторити  i = 0, i < 3 //так  //повторити  j = 0, j <= 0 //так  якщо first\_j\_in\_row – 1 >= 0 //ні  first\_j\_in\_row—  якщо last\_j\_in\_row + 1 < size (3) //так  last\_j\_in\_row++  .  .  .  На кожному кроці перевіряється наступний елемент по сходах. На першій ітерації змінній max присвоюється значення -88. На наступному береться значення -32. Воно більше за -88, тому max вже дорівнює -32. Далі -35 (менше -32, тому max не зміюється). Наступний крок: 91. Так, воно більше за -32, тому max дорівнює 91. -41 та 28 менші за 91, тому [0][0] елемент новой матриці дорівнює 91.  На наступному заході у функцію max\_in\_zone перевіряються елементи, які знаходяться по центру і все зводиться до таких порівнянь:  max = 49  => 49 > -32  => 49 > -35  => 49 > -10  => 49 < 91 (!) max = 91  => 91 > -41  => 91 > -32  => 91 > 28  matrix[0][1] = 91    На наступному заході у функцію max\_in\_zone перевіряються елементи, які знаходяться по лівій діагоналі все зводиться до таких порівнянь:  max = 98  => 98 > -35  => 98 > -10  => 98 < 91  => 98 > -41  => 98 > 28  matrix[0][2] = 98  За такою логікою порівнюються наступні ітерації, зменшуючи кількість рядків для пошуку на 1. |
|  | **кінець** |

1. ***Висновок***

*Під час виконання лабораторної роботи були набуті навички щодо використання двовимірних динамічних масивів, підпрограм користувача, роботи вказівників. Була використана стандартна функція C++ setw() для гарного виводу отриманої матриці на консоль користувачеві. Також використав функції rand() – для генерування випадкових чисел у заданому діапазоні. Також були використані багато арифметичних циклів та операторів альтернативного вибору.*