# 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

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження алгоритмів розгалуження»

Варіант 7

Виконав студент ІП-11 Головня Олександр Ростиславович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2021

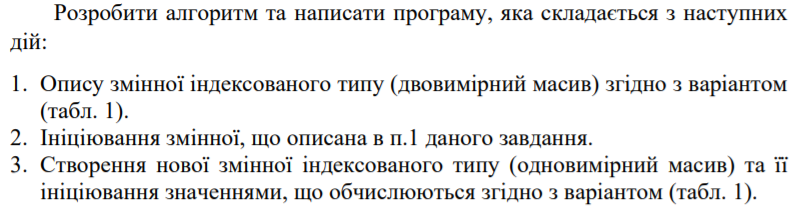
**Лабораторна робота № 8**

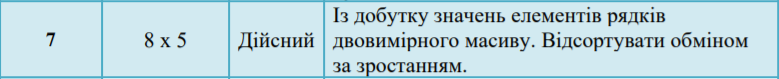
**Дослідження алгоритмів пошуку та сортування**

**Мета** – дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

**Індивідуальне завдання:**

Варіант 7.





**Постановка задачі**

Генеруємо двовимірний масив, ініціалізація одновимірного масиву одиничками(це потрібно для знаходження добутку). Присвоюємо значення добутку рядків двовимірного масиву елементам одновимірного. Сортуємо одновимірний масив методом обміну(Він же метод бульбашки)

**Побудова математичної моделі**

Складемо таблицю змінних

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| 1 | Дійсний | Dvomass | Початкове дане |
| 2 | Дійсний | tmp | Проміжне дане |
| 3 | Дійсний | mass | Вихідні дані |

**Розв’язання**

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії:

Крок 2. Деталізація заповнення першого масиву

Крок 3. Виведення першого масиву

Крок 4. Деталізація ініціалізації другого масиву

Крок 5. Виведення другого масиву

Крок 6. Сортировка та виведення другого посортированого масиву

**Псевдокод**

Крок 1

**Початок**

Деталізація заповнення першого масиву

Виведення першого масиву

Деталізація ініціалізації другого масиву

Виведення другого масиву

Сортировка та виведення другого посортированого масиву

**Кінець**

Крок 2

**Початок**

**Для (**i = 0; i < 8; i++) (рядок)

Для (j = 0; j < 5; j++) (Стовпець)

Dvomass[i][j] = rand() % 5 + 1;

Виведення першого масиву

Деталізація ініціалізації другого масиву

Виведення другого масиву

Сортировка та виведення другого посортированого масиву

**Кінець**

Крок 3

**Початок**

Для **(**i = 0; i < 8; i++) (рядок)

Для (j = 0; j < 5; j++) (Стовпець)

Dvomass[i][j] = rand() % 5 + 1;

Для **(**i = 0; i < 8; i++)

Для (j = 0; j < 5; j++)

Виведення Dvomass[i][j]

Деталізація ініціалізації другого масиву

Виведення другого масиву

Сортировка та виведення другого посортированого масиву

**Кінець**

Крок 4

**Початок**

Для **(**i = 0; i < 8; i++) (рядок)

Для (j = 0; j < 5; j++) (Стовпець)

Dvomass[i][j] = rand() % 5 + 1;

Для **(**i = 0; i < 8; i++)

Для (j = 0; j < 5; j++)

Виведення Dvomass[i][j]

Для (i = 0; i < 8; i++)

mass[i] = 1;

Для **(**i = 0; i < 8; i++)

Для (j = 0; j < 5; j++)

mass[i] = mass[i] \* Dvomass[i][j];

Виведення другого масиву

Сортировка та виведення другого посортированого масиву

**Кінець**

Крок 5

**Початок**

Для **(**i = 0; i < 8; i++) (рядок)

Для (j = 0; j < 5; j++) (Стовпець)

Dvomass[i][j] = rand() % 5 + 1;

Для **(**i = 0; i < 8; i++)

Для (j = 0; j < 5; j++)

Виведення Dvomass[i][j]

Для (i = 0; i < 8; i++)

mass[i] = 1;

Для **(**i = 0; i < 8; i++)

Для (j = 0; j < 5; j++)

mass[i] = mass[i] \* Dvomass[i][j];

Для (i = 0; i < 5; i++)

Виведення mass[i]

Сортировка та виведення другого посортированого масиву

**Кінець**

Крок 6

**Початок**

Для **(**i = 0; i < 8; i++) (рядок)

Для (j = 0; j < 5; j++) (Стовпець)

Dvomass[i][j] = rand() % 5 + 1;

Для **(**i = 0; i < 8; i++)

Для (j = 0; j < 5; j++)

Виведення Dvomass[i][j]

Для (i = 0; i < 8; i++)

mass[i] = 1;

Для **(**i = 0; i < 8; i++)

Для (j = 0; j < 5; j++)

mass[i] = mass[i] \* Dvomass[i][j];

Для (i = 0; i < 5; i++)

Виведення mass[i]

Для (i = 0; i < 8; i++)

Для (j = 8 - 1; j > i; j--)

Якщо (mass[j] < mass[j - 1])

tmp = mass[j];

mass[j] = mass[j - 1];

mass[j - 1] = tmp;

Виведення mass[i]

**Кінець**

**Блок-схема**

**Крок 1**



**Крок 2**



**Крок 3**



**Крок 4**

**Крок 5**

**Крок 6**



**Код програми**

#include<iostream>

#include<time.h>

using namespace std;

int main() {

double Dvomass[8][5]; //1 масив

srand(time(NULL));

for (int i = 0; i < 8; i++) //Заповнення 1 масиву

for (int j = 0; j < 5; j++)

Dvomass[i][j] = rand() % 5 + 1;

for (int i = 0; i < 8; i++) { //Виведення 1 масиву

for (int j = 0; j < 5; j++)

cout << Dvomass[i][j] << "\t";

cout << endl;

}

double mass[8]; //Ініціалізація 2 масиву одиничками

for (int i = 0; i < 8; i++)

mass[i] = 1;

for (int i = 0; i < 8; i++) { //Ініціалізація добутком двомасиву

for (int j = 0; j < 5; j++) {

mass[i] = mass[i] \* Dvomass[i][j];

}

}

cout << "massiv :" << endl;

for (int i = 0; i < 8; i++) //Виведення 2 масиву

cout << mass[i] << "\t";

cout << endl;

cout << "sort massiv :" << endl ;

//Сортировка методом бульбашки

for (int i = 0; i < 8; i++)

{

for (int j = 8 - 1; j > i; j--)

if (mass[j] < mass[j - 1])

{

int tmp = mass[j]; //tmp - доп змінна

mass[j] = mass[j - 1]; //Заміна числа на попереднє

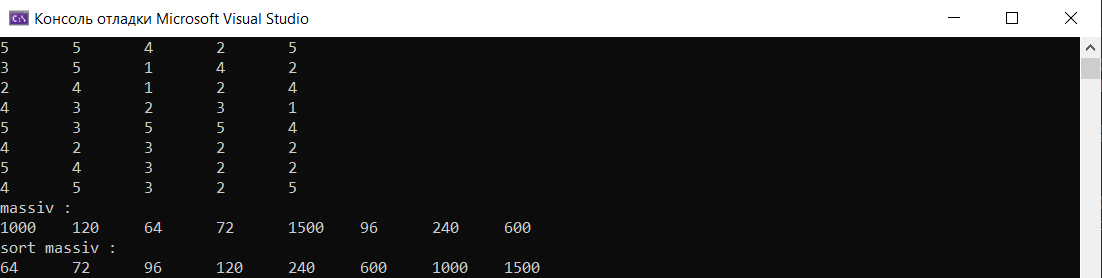
mass[j - 1] = tmp;

}

cout << mass[i] << "\t"; //виведення

}

}



**Випробування**

**Нехай перший рядок буде 5 5 4 2 5**

**Нехай другий рядок буде 3 5 1 4 2**

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| Добуток 1 | 5\*5\*4\*2\*5 = 1000 |
| Добуток 2 | 3\*5\*1\*4\*2 = 120 |

**Що і бачимо у виконнані програми. Сортування також проходить правильно, від мін до мах.**

**Висновок**

Отже, ми дослідили алгоритми пошуку та сортування, набули практичних навичок використання цих алгоритмів під час складання програмних специфікацій, а саме дослідив лінійний пошук невпорядкованого алгоритму, розробив програму що шукає добуток елементів кожного рядку та сортує масив методом обміну.