

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

про виконання лабораторних та практичних робіт блоку № 5

На тему: «Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли.
Стандартна бібліотека та деталі/методи роботи з файлами. Створення й
використання бібліотек.»

з дисципліни: «Основи програмування»

до:

ВНС Лабораторної Роботи № 6

ВНС Лабораторної Роботи № 8

ВНС Лабораторної Роботи № 9

Алготестер Лабораторної Роботи №4

Алготестер Лабораторної Роботи №6

Практичних Робіт до блоку №5

Виконав:

Студент групи ІІІ-12
Стик Назарій Олегович

Львів 2024

Тема роботи:

Файли. Бінарні Файли. Символи і Рядкові Змінні та Текстові Файли. Стандартна бібліотека та деталі/методи роботи з файлами. Створення й використання бібліотек.

Мета роботи:

Ознайомлення з основами роботи з файлами в мовах програмування, зокрема з бінарними та текстовими файлами. Вивчення основних операцій з файлами, таких як відкриття, читання, запис та закриття файлів. Ознайомлення з роботою зі символами та рядковими змінними при взаємодії з текстовими файлами. Детальне вивчення стандартної бібліотеки для роботи з файлами, а також методів для маніпулювання вмістом файлів. Окрему увагу буде приділено створенню та використанню бібліотек для забезпечення ефективності роботи з файлами та спрощення програмних рішень.

Теоретичні відомості:

У даній роботі розглядаються основи роботи з файлами в програмуванні, зокрема з текстовими та бінарними файлами. Текстові файли містять дані у вигляді тексту, що дозволяє зручно працювати з рядками та символами, тоді як бінарні файли використовуються для зберігання даних у форматі, який більш ефективний з точки зору обсягу пам'яті та швидкості доступу. Вивчаються методи роботи з файлами за допомогою стандартної бібліотеки мови програмування, зокрема функції для відкриття, читання, запису та закриття файлів. Особливу увагу приділено маніпулюванню рядковими змінними при роботі з текстовими файлами та використанню бінарних файлів для зберігання і передачі складних типів даних. Досліджується також створення та використання бібліотек для організації коду і спрощення роботи з файлами, що дозволяє підвищити ефективність програм та зменшити можливі помилки в процесі роботи з даними.

Джерела:

- Декілька відео на YouTube:
 - Файли на C++ - <https://www.youtube.com/watch?v=Cz4fl-TUjVk>
 - Бінарні Файли на C++ - <https://www.youtube.com/watch?v=fCvJ9Rsfy6c>
 - Створення й використання бібліотек на C++ - <https://www.youtube.com/watch?v=Wt4dxDNmDA8>
- Певну інформацію брав на сайтах:
 - <https://acode.com.ua/urok-220-bazovyj-fajlovyj-vvid-i-vyvid/>

- <https://acode.com.ua/urok-38-symvolnyi-typ-danyh-char/>
- https://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/66_C++/index.html

- Також вивчив багато інформації за допомогою ChatGPT.

Виконання роботи:

- **Завдання №1:** Theory Education Activities

Очікувано часу: **4 дні.**

Витрачено часу: **3 дні.**

- **Завдання №2:** Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7)

Очікувано часу: **1 година.**

Витрачено часу: **1 година.**

- **Завдання №3:** Lab# programming: VNS Lab 6 (Variant 19)

Код:

```

1  // Variant 19
2  // 19. Знищити з рядка всі слова, які не є ідентифікаторами.
3  #include <iostream>
4  #include <string>
5
6  using namespace std;
7
8  int main()
9  {
10     string line;
11     cout << "Enter a line: ";
12     cin >> line;
13     string word = "Hello";
14
15     for (int i = 0; i < line.length(); i++)
16     {
17         if (line[i] == ' ')
18         {
19             if (word[0] >= 'a' && word[0] <= 'z' || word[0] >= 'A' && word[0] <= 'Z')
20             {
21                 cout << word << " ";
22             }
23             word = "";
24         }
25         else
26         {
27             word += line[i];
28         }
29     }
30     if (word[0] >= 'a' && word[0] <= 'z' || word[0] >= 'A' && word[0] <= 'Z')
31     {
32         cout << word << " ";
33     }
34
35     return 0;
36 }
```

Очікувано часу: **30 хвилин.**

Витрачено часу: **1 година.**

- **Завдання №4:** Lab# programming: VNS Lab 8 (Variant 19)

Код:

```
1 // Variant 19
2 #include <iostream>
3
4 using namespace std;
5
6 struct Film
7 {
8     string name;
9     string director;
10    string country;
11    double profit;
12 };
13
14 int main()
15 {
16     Film films[3] = {
17         {"The Shawshank Redemption", "Frank Darabont", "USA", 58.3},
18         {"The Godfather", "Francis Ford Coppola", "USA", 246},
19         {"The Dark Knight", "Christopher Nolan", "USA", 1000}};
20     string name = "The Godfather";
21     Film newFilm = {"The Godfather: Part II", "Francis Ford Coppola", "USA", 48};
22     int n = 3;
23
24     for (int i = 0; i < n; i++)
25     {
26         cout << films[i].name << " " << films[i].director << " " << films[i].country << " " << films[i].profit << endl;
27     }
28     cout << endl;
29
30     n -= 2;
31     films[n] = newFilm;
32     for (int i = 0; i < n; i++)
33     {
34         cout << films[i].name << " " << films[i].director << " " << films[i].country << " " << films[i].profit << endl;
35     }
36
37
38
39
40     return 0;
41 }
42 // 19. Структура "Фільм":
43 // - назва;
44 // - режисер;
45 // - країна;
46 // - прибуток.
47 // Знищити 2 елементи з кінця файлу, додати елемент після елемента із зазначеною назвою.
```

Очікувано часу: **30 хвилин.**

Витрачено часу: **1,5 години.**

- **Завдання №5:** Lab# programming: VNS Lab 9 (Variant 19)

Код:

```

1 // Variant 19
2 #include <iostream>
3 #include <fstream>
4 #include <string>
5
6 using namespace std;
7
8 // Функція для перевірки, чи є символ приголосним
9 bool isConsonant(char c)
10 {
11     c = tolower(c); // Переводимо в нижній регістр
12     return (c >= 'a' && c <= 'z') && !(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');
13 }
14
15 int main()
16 {
17     ifstream fileF1("F1.txt"); // Вхідний файл
18     ofstream fileF2("F2.txt"); // Вихідний файл
19
20     if (!fileF1.is_open() || !fileF2.is_open())
21     {
22         cout << "Не вдалося відкрити файли!" << endl;
23         return 1;
24     }
25
26     string line;
27     string firstWord;
28
29     // Зчитуємо перший рядок і отримуємо перше слово
30     if (getline(fileF1, line))
31     {
32         size_t pos = line.find(' ');
33         if (pos != string::npos)
34         {
35             firstWord = line.substr(0, pos); // Зберігаємо перше слово
36         }
37         else
38         {
39             firstWord = line; // Якщо в рядку лише одне слово
40         }
41     }
42
43     // Читаємо інші рядки та записуємо в F2, якщо немає співпадіння з першим словом
44     while (getline(fileF1, line))
45     {
46         bool containsFirstWord = false;
47         size_t pos = 0;
48
49         // Шукаємо слова в рядку
50         while ((pos = line.find(' ', pos)) != string::npos)
51         {
52             string word = line.substr(0, pos);
53             if (word == firstWord)
54             {
55                 containsFirstWord = true;
56                 break;
57             }
58             line.erase(0, pos + 1);
59         }
60
61         // Якщо не містить перше слово, записуємо рядок у F2
62         if (!containsFirstWord)
63         {
64             fileF2 << line << endl;
65         }
66     }
67
68     // Закриваємо файли
69     fileF1.close();
70     fileF2.close();
71
72     // Відкриваємо файл F2 для підрахунку приголосних
73     ifstream fileF2ForCounting("F2.txt");
74
75     if (!fileF2ForCounting.is_open()) {
76         cout << "Не вдалося відкрити файл F2 для підрахунку приголосних!" << endl;
77         return 1;
78     }
79
80     int consonantCount = 0;
81     // Зчитуємо перший рядок з F2
82     if (getline(fileF2ForCounting, line))
83     {
84         for (char c : line)
85         {
86             if (isConsonant(c))
87             {
88                 consonantCount++;
89             }
90         }
91     }
92
93     cout << "Кількість приголосних в першому рядку файлу F2: " << consonantCount << endl;
94     fileF2ForCounting.close();
95
96     return 0;
97 }

```

```

79
80 int consonantCount = 0;
81 // Зчитуємо перший рядок з F2
82 if (getline(fileF2ForCounting, line))
83 {
84     for (char c : line)
85     {
86         if (isConsonant(c))
87         {
88             consonantCount++;
89         }
90     }
91 }
92
93 cout << "Кількість приголосних в першому рядку файлу F2: " << consonantCount << endl;
94 fileF2ForCounting.close();
95
96 return 0;
97 }
98
99
100 // 1) Скопіювати з файлу F1 у файл F2 всі рядки, у яких немає слів, що збігаються з першим словом.
101 // 2) Визначити кількість приголосних букв у першому рядку файлу F2.

```

Очікувано часу: 1 година.

Витрачено часу: 2 години.

- Завдання №6: Lab# programming: Algotester Lab 4

Файл 4.1 (варант-1)

Код:

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  #include <set>
5
6  int main()
7  {
8      int N, M;
9
10     // Введення першого масиву
11     std::cin >> N;
12     std::vector<int> array1(N);
13     for (int i = 0; i < N; ++i)
14     {
15         std::cin >> array1[i];
16     }
17
18     // Введення другого масиву
19     std::cin >> M;
20     std::vector<int> array2(M);
21     for (int i = 0; i < M; ++i)
22     {
23         std::cin >> array2[i];
24     }
25
26     // Сортують обидва масиви для правильного виконання операцій STL
27     std::sort(array1.begin(), array1.end());
28     std::sort(array2.begin(), array2.end());
29
30     std::vector<int> result;
31
32     // 1. Різниця N-M
33     std::set_difference(array1.begin(), array1.end(), array2.begin(), array2.end(), std::back_inserter(result));
34     std::cout << result.size() << "\n";
35     for (int num : result) std::cout << num << " ";
36     std::cout << "\n";
37     result.clear();
38
39     // 2. Різниця M-N
40     std::set_difference(array2.begin(), array2.end(), array1.begin(), array1.end(), std::back_inserter(result));
41     std::cout << result.size() << "\n";
42     for (int num : result) std::cout << num << " ";
43     std::cout << "\n";
44     result.clear();
45
46     // 3. Перетин
47     std::set_intersection(array1.begin(), array1.end(), array2.begin(), array2.end(), std::back_inserter(result));
48     std::cout << result.size() << "\n";
49     for (int num : result) std::cout << num << " ";
50     std::cout << "\n";
51     result.clear();
52
53     // 4. Об'єднання
54     std::set_union(array1.begin(), array1.end(), array2.begin(), array2.end(), std::back_inserter(result));
55     std::cout << result.size() << "\n";
56     for (int num : result) std::cout << num << " ";
57     std::cout << "\n";
58     result.clear();
59
60     // 5. Симетрична різниця
61     std::set_symmetric_difference(array1.begin(), array1.end(), array2.begin(), array2.end(), std::back_inserter(result));
62     std::cout << result.size() << "\n";
63     for (int num : result) std::cout << num << " ";
64     std::cout << "\n";
65
66     return 0;
67 }
68
69 // Ваше завдання вивести:
70 // 1. Різницю N-M
71 // 2. Різницю M-N
72 // 3. Їх перетин
73 // 4. Їх об'єднання
74 // 5. Їх симетричну різницю
```

Очікувано часу: **40 хвилин.**

Витрачено часу: **2 години.**

Файл 4.2 (варант-3)

Код:

```
1  #include <iostream>
2
3  void customSort(int arr[], int size, bool ascending)
4  {
5      // Просте сортування вставкою
6      for (int i = 1; i < size; i++)
7      {
8          int key = arr[i];
9          int j = i - 1;
10         if (ascending)
11         {
12             // Сортування за зростанням
13             while (j >= 0 && arr[j] > key)
14             {
15                 arr[j + 1] = arr[j];
16                 j--;
17             }
18         }
19         else
20         {
21             // Сортування за спаданням
22             while (j >= 0 && arr[j] < key)
23             {
24                 arr[j + 1] = arr[j];
25                 j--;
26             }
27         }
28         arr[j + 1] = key;
29     }
30 }
31
32 int main()
33 {
34     int n;
35     std::cin >> n;
36
37     int arr[1000];
38     for (int i = 0; i < n; ++i)
39     {
40         std::cin >> arr[i];
41     }
42
43     // Масиви для збереження чисел з різними остачами
44     int mod0[1000], mod1[1000], mod2[1000];
45     int count0 = 0, count1 = 0, count2 = 0;
46
47     // Розподіл чисел на три групи за остачею
```

```
48     for (int i = 0; i < n; ++i)
49     {
50         if (arr[i] % 3 == 0)
51         {
52             mod0[count0++] = arr[i];
53         }
54         else if (arr[i] % 3 == 1)
55         {
56             mod1[count1++] = arr[i];
57         }
58         else
59         {
60             mod2[count2++] = arr[i];
61         }
62     }
63
64     // Сортування груп за умовами
65     customSort(mod0, count0, true); // Числа з остачею 0 за зростанням
66     customSort(mod1, count1, false); // Числа з остачею 1 за спаданням
67     customSort(mod2, count2, true); // Числа з остачею 2 за зростанням
68
69     // Об'єднання масивів у один та видалення дублікатів
70     int result[1000];
71     int resultSize = 0;
72
73     auto addUnique = [&](int value)
74     {
75         for (int i = 0; i < resultSize; ++i)
76         {
77             if (result[i] == value) return; // Якщо значення вже існує, не додаємо
78         }
79         result[resultSize++] = value;
80     };
81
82     for (int i = 0; i < count0; ++i)
83     {
84         addUnique(mod0[i]);
85     }
86     for (int i = 0; i < count1; ++i)
87     {
88         addUnique(mod1[i]);
89     }
90     for (int i = 0; i < count2; ++i)
91     {
```

```

90     for (int i = 0; i < count2; ++i)
91     {
92         addUnique(mod2[i]);
93     }
94
95     // Виведення результату
96     std::cout << resultSize << "\n";
97     for (int i = 0; i < resultSize; ++i)
98     {
99         std::cout << result[i] << " ";
100     }
101     std::cout << "\n";
102
103     return 0;
104 }
105

```

Очікувано часу: **30 хвилин.**

Витрачено часу: **1 година.**

- **Завдання №7:** Lab# programming: Algotester Lab 6 (Variant-3)

Код:

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main()
7  {
8      int N, Q;
9      cin >> N;
10
11      // Ініціалізація матриці
12      vector<vector<int>> grid(N, vector<int>(N));
13      for (int i = 0; i < N; i++)
14      {
15          for (int j = 0; j < N; j++)
16          {
17              char ch;
18              cin >> ch;
19              grid[i][j] = ch - '0'; // Перетворення символу на цифру
20          }
21      }
22
23      // Зберігаємо інформацію про використані числа для кожного рядка та стовпця
24      vector<vector<bool>> usedInRow(N, vector<bool>(N + 1, false));
25      vector<vector<bool>> usedInCol(N, vector<bool>(N + 1, false));
26
27      // Заповнюємо дані про використані числа
28      for (int i = 0; i < N; i++)
29      {
30          for (int j = 0; j < N; j++)
31          {
32              int val = grid[i][j];
33              if (val != 0)
34              {
35                  usedInRow[i][val] = true;
36                  usedInCol[j][val] = true;
37              }
38          }
39      }
40
41      // Зчитуємо кількість запитів
42      cin >> Q;
43      vector<pair<int, int>> queries(Q);
44      for (int i = 0; i < Q; i++)
45      {
46          cin >> queries[i].first >> queries[i].second;
47          queries[i].first--; // Перетворення до 0-індексації
48          queries[i].second--;
49      }
50
51      // Обробка кожного запиту
52      for (int i = 0; i < Q; i++)
53      {
54          int x = queries[i].first;
55          int y = queries[i].second;
56
57          if (grid[x][y] != 0)
58          {
59              // Якщо клітинка вже заповнена, виводимо 1 і значення клітинки
60              cout << "1 " << grid[x][y] << endl;
61              continue;
62          }
63

```



```

63
64 // Формування списку можливих чисел на основі 'usedInRow' та 'usedInCol'
65 vector<int> possibleNumbers;
66 for (int num = 1; num <= N; num++)
67 {
68     if (!usedInRow[x][num] && !usedInCol[y][num])
69     {
70         possibleNumbers.push_back(num);
71     }
72 }
73
74 // Виводимо результат для поточного запиту
75 if (possibleNumbers.empty())
76 {
77     cout << "0" << endl;
78 }
79 else
80 {
81     cout << possibleNumbers.size() << " ";
82     for (int k = 0; k < possibleNumbers.size(); k++)
83     {
84         cout << possibleNumbers[k];
85         if (k < possibleNumbers.size() - 1)
86         {
87             cout << " ";
88         }
89     }
90     cout << endl;
91 }
92 }
93
94 return 0;
95
96

```

Очікувано часу: **30 хвилин.**

Витрачено часу: **1,5 години.**

- Завдання №8: Practice# programming: Class Practice Task

Код(Завдання 1):

```

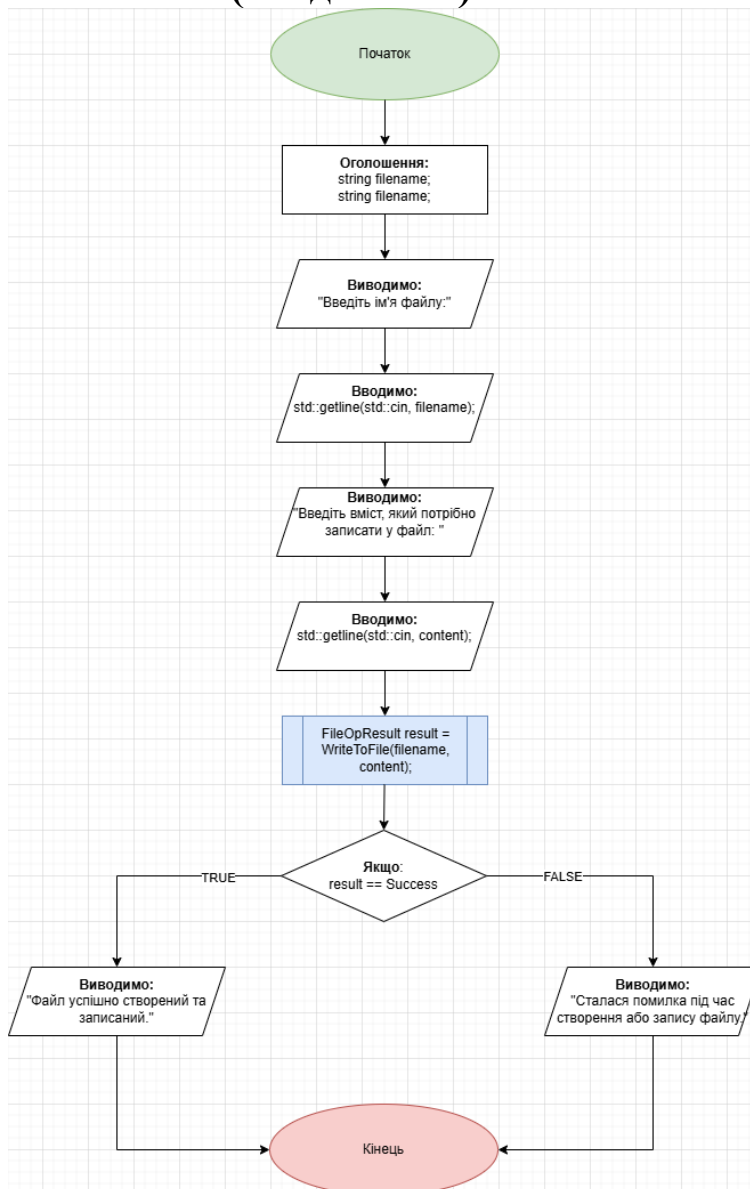
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  enum FileOpResult { Success, Failure };
6
7  FileOpResult WriteToFile(const std::string& name, const std::string& content)
8  {
9      // Перевірка на допустимість імені файлу та контенту
10     if (name.empty() || content.empty())
11     {
12         return Failure;
13     }
14
15     std::ofstream outFile(name);
16
17     // Перевірка, чи файл відкрився успішно
18     if (!outFile.is_open())
19     {
20         return Failure;
21     }
22
23     outFile << content;
24
25     // Перевірка на успішність запису
26     if (outFile.fail())
27     {
28         outFile.close();
29         return Failure;
30     }
31
32     outFile.close();
33
34     return Success;
35 }
36
37 int main()
38 {
39     std::string filename;
40     std::string content;
41
42     std::cout << "Введіть ім'я файлу: ";
43     std::getline(std::cin, filename);
44
45     std::cout << "Введіть вміст, який потрібно записати в файл: ";
46     std::getline(std::cin, content);
47
48     FileOpResult result = WriteToFile(filename, content);
49
50     if (result == Success)
51     {
52         std::cout << "Файл успішно створений та записаний.\n";
53     }
54     else
55     {
56         std::cout << "Сталася помилка під час створення або запису файлу.\n";
57     }
58
59     return 0;
60 }

```

Очікувано часу: 1 година.

Витрачено часу: 30 хвилин.

Блок-схема(Завдання 1):



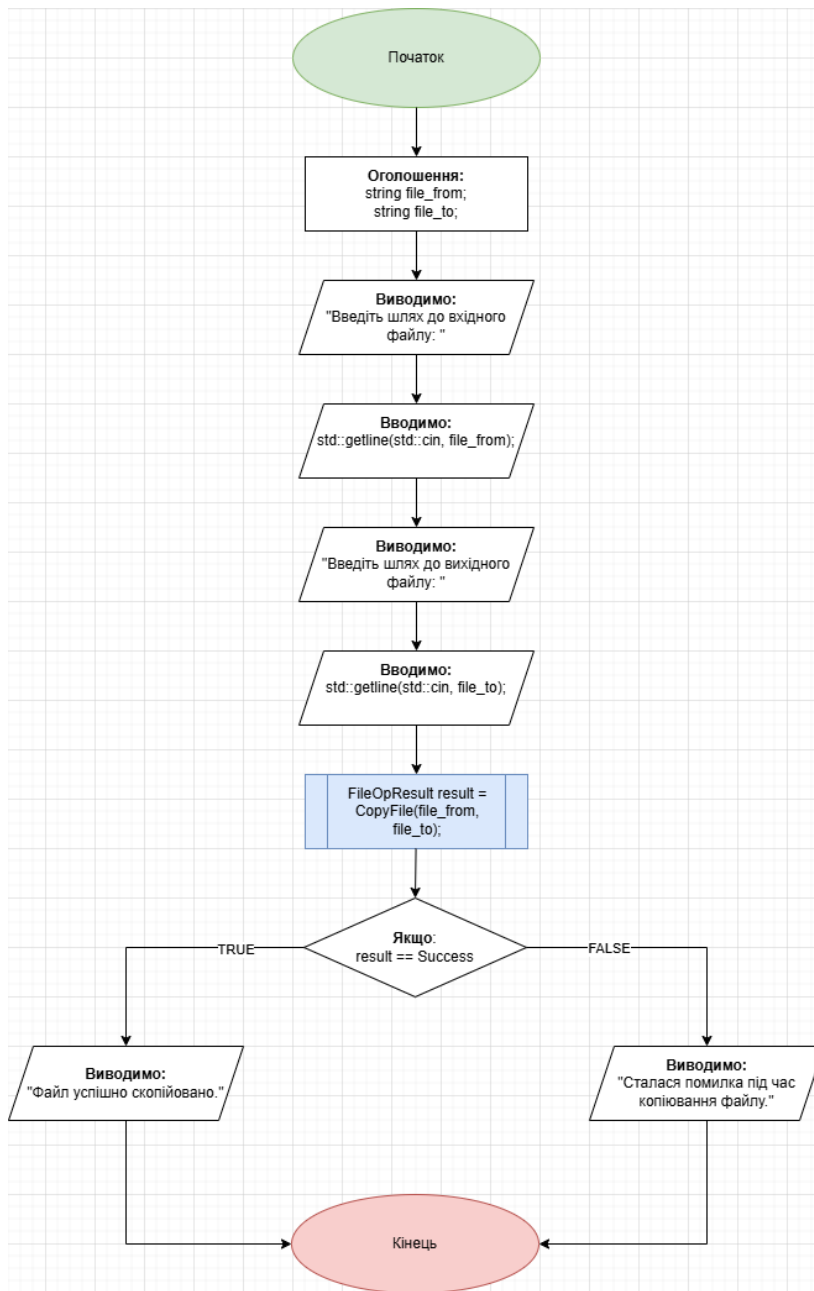
Код(Завдання 2):

```
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  enum FileOpResult { Success, Failure };
6
7  FileOpResult CopyFile(const std::string& file_from, const std::string& file_to)
8  {
9      // Відкриваємо вхідний файл для читання
10     std::ifstream inFile(file_from, std::ios::in | std::ios::binary);
11
12     if (!inFile.is_open())
13     {
14         std::cerr << "Не вдалося відкрити файл для читання: " << file_from << std::endl;
15         return Failure;
16     }
17
18     // Відкриваємо вихідний файл для запису
19     std::ofstream outFile(file_to, std::ios::out | std::ios::binary);
20
21     if (!outFile.is_open())
22     {
23         std::cerr << "Не вдалося відкрити файл для запису: " << file_to << std::endl;
24         inFile.close();
25         return Failure;
26     }
27
28     outFile << inFile.rdbuf();
29
30     if (outFile.fail())
31     {
32         std::cerr << "Сталася помилка під час запису в файл: " << file_to << std::endl;
33         inFile.close();
34         outFile.close();
35         return Failure;
36     }
37
38     inFile.close();
39     outFile.close();
40
41     return Success;
42 }
43
44 int main()
45 {
46     std::string file_from, file_to;
47
48     std::cout << "Введіть шлях до вхідного файлу: ";
49     std::getline(std::cin, file_from);
50
51     std::cout << "Введіть шлях до вихідного файлу: ";
52     std::getline(std::cin, file_to);
53
54     FileOpResult result = CopyFile(file_from, file_to);
55
56     if (result == Success)
57     {
58         std::cout << "Файл успішно скопійовано.\n";
59     }
60     else
61     {
62         std::cout << "Сталася помилка під час копіювання файлу.\n";
63     }
64
65     return 0;
66 }
67
```

Очікувано часу: 30 хвилин.

Витрачено часу: 45 хвилин.

Блок-схема(Завдання 2):



- **Завдання №9:** Practice# programming: Self Practice Task

Код:

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main()
6  {
7      int n, k;
8      cin >> n >> k;
9      vector<int> a(n);
10     for (int i = 0; i < n; i++)
11     {
12         cin >> a[i];
13     }
14
15     int max_length = 0, current_length = 0;
16
17     for (int i = 0; i < n; i++)
18     {
19         if (a[i] >= k)
20         {
21             current_length++;
22             max_length = max(max_length, current_length);
23         }
24         else
25         {
26             current_length = 0;
27         }
28     }
29
30     cout << max_length << endl;
31     return 0;
32 }
```

Очікувано часу: **30 хвилин.**

Витрачено часу: **20 хвилин.**

Pull-Request:

Висновок: У ході цієї роботи я ознайомився з основами роботи з файлами в мовах програмування, зокрема з бінарними та текстовими файлами. Я вивчив основні операції, такі як відкриття, читання, запис та закриття файлів, а також особливості роботи зі символами та рядковими змінними при взаємодії з текстовими файлами. Особливу увагу я приділив вивченню стандартної бібліотеки для роботи з файлами, а також методам для маніпулювання їхнім вмістом. Крім того, я ознайомився з процесом створення та використання бібліотек, що значно підвищують ефективність роботи з файлами та спрощують розробку програмних рішень.