

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 ШІ-11
Вербицький Юрій Віталійович

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: Ознайомитися з принципами роботи з файлами в програмуванні, зокрема бінарними та текстовими файлами. Навчитися використовувати символи, рядкові змінні та текстові файли у програмах. Розглянути функціонал стандартної бібліотеки для роботи з файлами, а також методи створення та використання власних бібліотек для роботи з файлами.

Теоретичні відомості:

1)Перелік тем:

1. Вступ до Роботи з Файлами;
2. Символи і Рядкові Змінні;
3. Текстові Файли;
4. Бінарні Файли;
5. Стандартна бібліотека та робота з файлами;
6. Створення й використання бібліотек.

2)Індивідуальний план опрацювання теорії:

1. Вступ до Роботи з Файлами

<https://acode.com.ua/urok-220-bazovyj-fajlovyj-vvid-i-vyvid/>

2. Символи і Рядкові Змінні

https://www.w3schools.com/cpp/cpp_files.asp

3. Текстові Файли

<https://www.geeksforgeeks.org/file-handling-c-classes/>

4. Бінарні Файли

<https://www.geeksforgeeks.org/difference-between-cpp-text-file-and-binary-file/>

5. Стандартна бібліотека та робота з файлами

- Unordered map: [unordered_map in C++ STL - GeeksforGeeks](https://www.geeksforgeeks.org/unordered-map-in-cpp-stl/)
- Стандарти бібліотеки і їх виклик:
https://en.cppreference.com/w/cpp/standard_library

6. Створення й використання бібліотек

<https://stackoverflow.com/questions/16693273/how-do-i-create-a-library>

<https://www.geeksforgeeks.org/how-do-i-create-a-library-in-cpp/>

Виконання роботи:

1)Перелік завдань:

- John Black - Epic 5 Task 1 - Theory Education Activities
- John Black - Epic 5 Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-9)
- John Black - Epic 5 Task 3 - Lab# programming: VNS Lab 6(варіант 4)
- John Black - Epic 5 Task 4 - Lab# programming: VNS Lab 8(варіант 4)
- John Black - Epic 5 Task 5 - Lab# programming: VNS Lab 9(варіант 4)
- John Black - Epic 5 Task 6 - Lab# programming: Algotester Lab 4(варіант 1)
- John Black - Epic 5 Task 7 - Lab# programming: Algotester Lab 6(варіант 1)
- John Black - Epic 5 Task 8 - Practice# programming: Class Practice Task
- John Black - Epic 5 Task 9 - Practice# programming: Self Practice Task
- John Black - Epic 5 Task 10 - Result Documentation Report and Outcomes Placement Activities (Docs and Programs on GitHub)
- John Black - Epic 5 Task 11 - Results Evaluation and Release

2)Умови завдань:

Task 3: Задано рядок, що складається із символів. Символи поєднуються в слова. Слова одне від одного відокремлюються одним або декількома пробілами. Наприкінці тексту ставиться крапка. Текст містить не більше 255 символів. Виконати ввід рядка, використовуючи функцію gets(s) і здійснити обробку рядка у відповідності зі своїм варіантом.

17. Всі слова рядка, які починаються з букви, відсортувати за абеткою.

Task 4: Сформувати двійковий файл із елементів, заданої у варіанті структури, роздрукувати його вміст, виконати знищення й додавання елементів у відповідності зі своїм варіантом, використовуючи для пошуку елементів що знищуються чи додаються, функцію. Формування, друк, додавання й знищення елементів оформити у вигляді функцій. Передбачити повідомлення про помилки при відкритті файлу й виконанні операцій вводу/виводу.

17. Структура "Фільм":

- назва;
- режисер;
- рік випуску;
- вартість.

Знищити елемент із заданим номером, додати 2 елементи перед елементом із заданим прізвищем.

Task 5: Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію

Виконати завдання:

17.

1) Скопіювати з файлу F1 у файл F2 всі рядки, що починаються на букву «А» розташовані між рядками з номерами N1 й N2, а потім всі рядки від N2+3 і до останнього.

2) Визначити кількість слів в останньому рядку файлу F2.

Task 6: Lab 4v2

Обмеження: 1 сек., 256 MiB

Вам дано масив aa з NN цілих чисел.

Спочатку видаліть масиву aa усі елементи що повторюються, наприклад масив [1, 3, 3, 4] має перетворитися у [1, 3, 4].

Після цього оберніть посортовану версію масиву aa на KK, тобто

при K=3K=3 масив [1, 2, 3, 4, 5, 6, 7] перетвориться на [4, 5, 6, 7, 1, 2, 3].

Виведіть результат.

Input

У першому рядку цілі числа NN та KK

У другому рядку NN цілих чисел - елементи масиву aa

Output

У першому рядку ціле число NN - розмір множини aa

У наступному рядку NN цілих чисел - множина a.

Task 7: Lab 6v3

Обмеження: 2 сек., 256 MiB

У Клінта в черговий раз виключилось світло і йому немає чим зайнятися. Так як навіть це не заставить його подивитися збережені відео про програмування на ютубі - він вирішив придумати свою гру на основі sudoku.

Гра виглядає так:

Є поле розміром $N \times N \times N$, в якому частина клітинок заповнена цифрами, а частина клітинок пусті (позначаються нулем). Також у нього є QQ пар координат XX та YY .

Завданням гри є написати до кожної координати скільки чисел туди можна вписати (якщо вона пуста) і які це числа (обов'язково в посортовані по зростанню!). В клітинку можна вписати лише ті числа, які не зустрічаються в рядку та стовбці, які перетинаються у цій клітинці.

Під час гри поле не міняється!

Також необов'язково, щоб це було валідне sudoku! Якщо є клітинка, в яку не можна вписати ніяку цифру - виведіть 0.

Також допускаються рядки та стовпці, в яких цифра записана кілька разів.

Input

У першому рядку ціле число NN - розмір поля для гри

У NN наступних рядках стрічка row_i яка складається з NN цифр - i -й рядок.

Ціле число QQ - кількість запитань

У наступних QQ рядках 2 цілих числа x_j, y_j - координати клітинок j -го запитання

Output

QQ разів відповідь у наступному форматі:

Натуральне число MM - кількість цифр, які можна вписати в клітинку

MM цифр розділених пробілом - можливі цифри

Task 8: Задача №1 – Запис текстової стрічки у файл із заданим ім'ям

Реалізувати функцію створення файла і запису в нього даних:

```
enum FileOpResult { Success, Failure, ... };
```

```
FileOpResult write_to_file(char *name, char *content);
```

Умови задачі:

- створити файл із заданим ім'ям; якщо файл існує – перезаписати його вміст
- написати код стійкий до різних варіантів вхідних параметрів
- name – ім'я, може не включати шлях
- записати у файл вміст стрічки content, прочитати content із стандартного вводу
- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, запису даних, чи закриття файла.

Задача №2 – Копіювання вмісту файла у інший файл

Реалізувати функцію створення файла і запису в нього даних:

```
enum FileOpResult { Success, Failure, ... };
```

```
FileOpResult copy_file(char *file_from, char *file_to);
```

Умови задачі:

- копіювати вміст файла з ім'ям file_from у файл з ім'ям file_to; написати код стійкий до різних варіантів вхідних параметрів, обробити всі можливі варіанти відсутності одного з файлів
- file_from, file_to – можуть бути повним або відносним шляхом
- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, читання чи запису даних, закриття файла.

Task 9: Завдання:

Вам необхідно розробити програму, яка виконує наступні дії:

Зчитує текстовий файл input.txt. Визначає кількість слів у файлі, кількість рядків та кількість символів (включаючи пробіли). Зберігає результати аналізу у файл output.txt у форматі:

Копіювати код

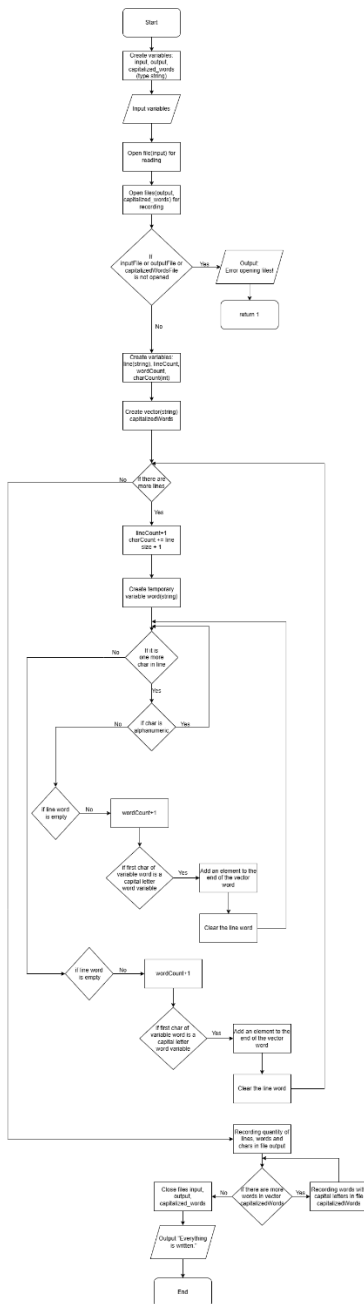
Кількість рядків: X

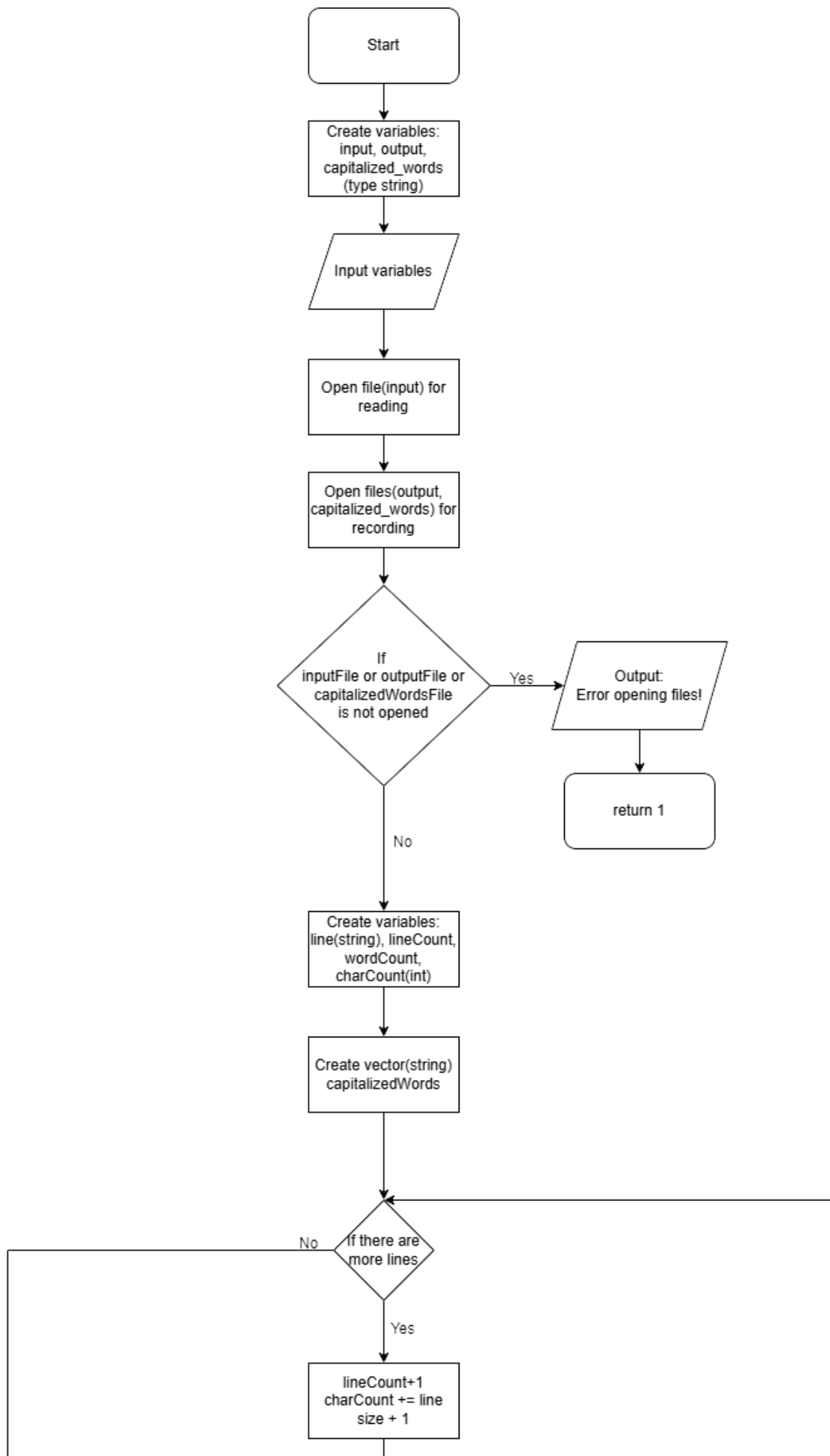
Кількість слів: Y

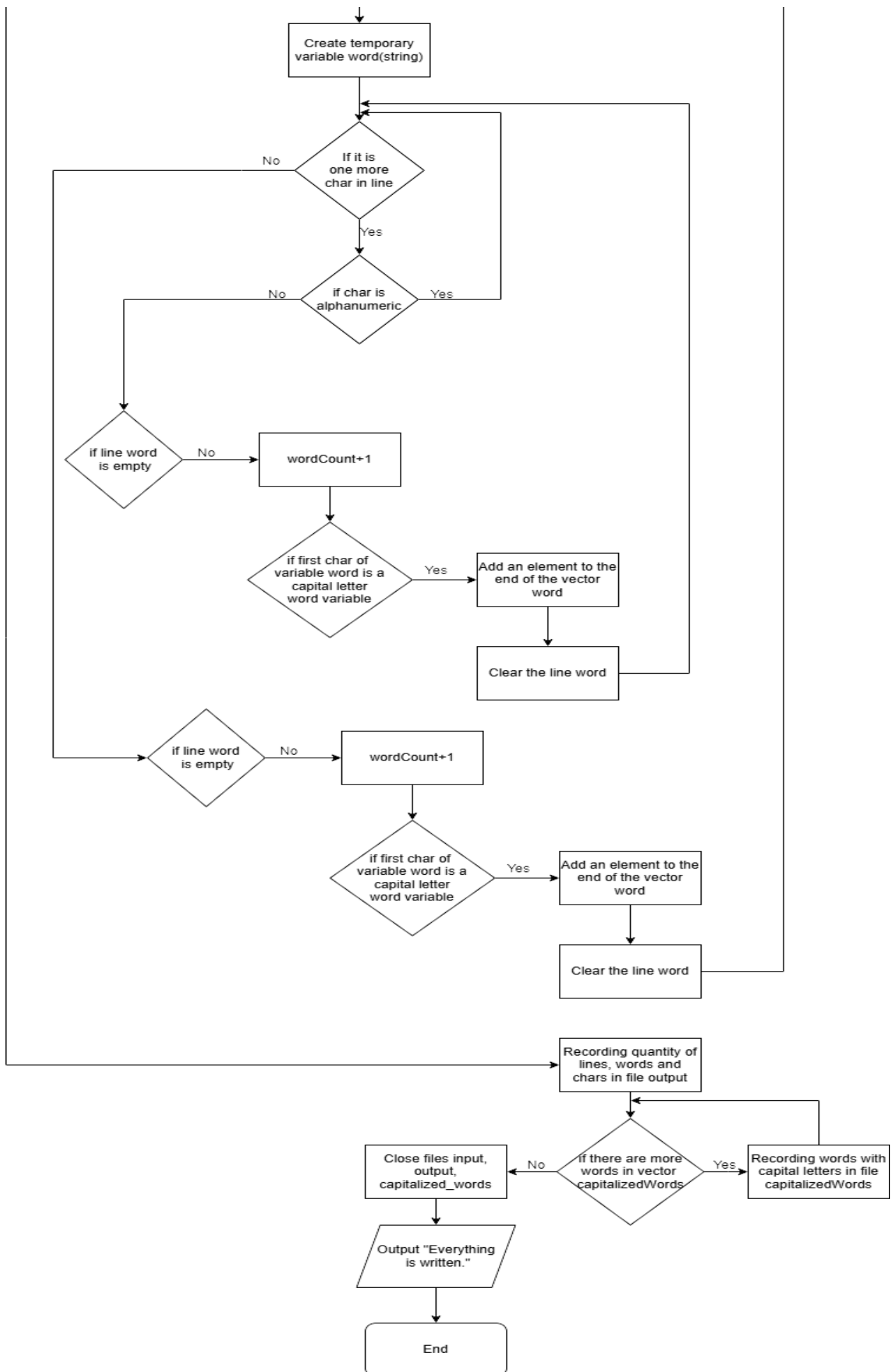
Кількість символів: Z

Знаходить усі слова, які починаються з великої літери, і записує їх у файл capitalized_words.txt, кожне слово з нового рядка.

Вхідні дані:







Орієнтовний час виконання: 1.5 год

4) Код програм з посиланням на зовнішні ресурси:

John Black - Epic 5 Task 3 - Lab# programming: VNS Lab 6(варіант 17)

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <ctype.h>
4
5  int firstletter(char *word){
6      return isalpha(word[0]);
7  }
8
9  int main(){
10
11     int maxlen = 255;
12     char str[maxlen + 1];
13     char *words[maxlen];
14     int k = 0;
15     printf("Enter line: \n");
16     gets(str);
17
18     char *token = strtok(str, " ");
19     while (token != NULL) {
20         if (firstletter(token)) {
21             words[k++] = token;
22         }
23         token = strtok(NULL, " ");
24     }
25
26     for (int i = 0; i < k - 1; i++) {
27         for (int j = i + 1; j < k; j++) {
28             if (strcmp(words[i], words[j]) > 0) {
29                 char *temp = words[i];
30                 words[i] = words[j];
31                 words[j] = temp;
32             }
33         }
34     }
35
36     printf("Sorted words:\n");
37     for (int i = 0; i < k; i++) {
38         printf("%s\n", words[i]);
39     }
40 }
41
42     return 0;
43 }
```

John Black - Epic 5 Task 4 - Lab# programming: VNS Lab 8(вариант 17)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  #define MAX_LEN 100
6
7  typedef struct {
8      char title[100];
9      char director[100];
10     int year;
11     float price;
12 } Film;
13
14 void createFilmFile(const char *filename) {
15     FILE *file = fopen(filename, "wb");
16     if (file == NULL) {
17         perror("Error: can't open file");
18         return;
19     }
20
21     Film films[] = {
22         {"The Godfather", "Francis Ford Coppola", 1972, 9.99},
23         {"The Shawshank Redemption", "Frank Darabont", 1994, 14.99},
24         {"The Dark Knight", "Christopher Nolan", 2008, 19.99}
25     };
26
27     size_t filmCount = sizeof(films) / sizeof(films[0]);
28     fwrite(films, sizeof(Film), filmCount, file);
29
30     fclose(file);
31 }
32
33 void addFilmBeforeDirector(const char *filename, const char *director, Film newFilm1, Film newFilm2) {
34     FILE *file = fopen(filename, "rb+");
35     if (file == NULL) {
36         perror("Error: can't open file");
37         return;
38     }
39
40     Film films[MAX_LEN];
41     size_t count = 0;
42
43     while (fread(&films[count], sizeof(Film), 1, file)) {
44         count++;
45     }
46
47     size_t i;
48     for (i = 0; i < count; i++) {
49         if (strcmp(films[i].director, director) == 0) {
50             break;
51         }
52     }
53
54     if (i < count) {
55         memmove(&films[i + 2], &films[i], sizeof(Film) * (count - i));
56
57         films[i] = newFilm1;
58         films[i + 1] = newFilm2;
59         count += 2;
60
61         freopen(filename, "wb", file);
62         fwrite(films, sizeof(Film), count + 2, file);
63     } else {
64         printf("Director not found.\n");
65     }
66
67     fclose(file);
68 }
69
70 void deleteFilmByIndex(const char *filename, size_t index) {
71     FILE *file = fopen(filename, "rb+");
72     if (file == NULL) {
73         perror("Error: can't open file");
74         return;
75     }
76
77     Film films[MAX_LEN];
```

```

78     size_t count = 0;
79
80     while (fread(&films[count], sizeof(Film), 1, file)) {
81         count++;
82     }
83
84     if (index >= count) {
85         printf("The movie with this index was not found.\n");
86         fclose(file);
87         return;
88     }
89
90     for (size_t i = index; i < count - 1; i++) {
91         films[i] = films[i + 1];
92     }
93     count--;
94
95     freopen(filename, "wb", file);
96     fwrite(films, sizeof(Film), count, file);
97
98     fclose(file);
99 }
100
101 void printFilms(const char *filename) {
102     FILE *file = fopen(filename, "rb");
103     if (file == NULL) {
104         perror("Error: can't open file");
105         return;
106     }
107
108     Film film;
109     while (fread(&film, sizeof(Film), 1, file)) {
110         if (strlen(film.title) == 0 && strlen(film.director) == 0 && film.year == 0 && film.price == 0) {
111             continue;
112         }
113         printf("Name: %s\n", film.title);
114         printf("Director: %s\n", film.director);
115         printf("Year of release: %d\n", film.year);
116         printf("Price: %.2f\n\n", film.price);
117     }
118     fclose(file);
119 }
120
121
122 int main() {
123     const char *filename = "films.dat";
124
125     createFilmFile(filename);
126
127     printf("Contents of the file before changes:\n");
128     printFilms(filename);
129
130     Film newFilm1 = {"Inception", "Christopher Nolan", 2010, 15.99};
131     Film newFilm2 = {"Memento", "Christopher Nolan", 2000, 12.99};
132     addFilmBeforeDirector(filename, "Christopher Nolan", newFilm1, newFilm2);
133
134     printf("Contents of the file after changes:\n");
135     printFilms(filename);
136
137     deleteFilmByIndex(filename, 1);
138
139     printf("The contents of the file after deleting the element:\n");
140     printFilms(filename);
141
142     return 0;
143 }

```

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <cctype>
5
6  using namespace std;
7
8  int countWords(const string &line) {
9      int count = 0, inWord = 0;
10     for (char ch : line) {
11         if (isspace(ch)) {
12             inWord = 0;
13         } else if (!inWord) {
14             inWord = 1;
15             count++;
16         }
17     }
18     return count;
19 }
20
21 void createFileF1(const char *filename) {
22     ofstream file(filename);
23     if (!file.is_open()) {
24         perror("Error creating file F1");
25         exit(1);
26     }
27
28     cout << "Enter 10 lines of text to write to file F1 (spaces are optional):\n";
29     for (int i = 0; i < 10; i++) {
30         string line;
31         getline(cin, line);
32         file << line << "\n";
33     }
34
35     file.close();
36 }
37
38 void copyToFileF2(const char *file1, const char *file2, int N1, int N2) {
39     ifstream f1(file1);
40     ofstream f2(file2);
41     if (!f1.is_open() || !f2.is_open()) {
42         perror("Error opening files");
43         exit(1);
44     }
45
46     string line;
47     int currentLine = 0;
48
49     while (getline(f1, line)) {
50         currentLine++;
51         if ((currentLine >= N1 && currentLine <= N2 && !line.empty() && line[0] == 'A') ||
52             currentLine > N2 + 2) {
53             f2 << line << "\n";
54         }
55     }
56
57     f1.close();
58     f2.close();
59 }
60
61 void printFile(const char *filename) {
62     ifstream file(filename);
63     if (!file.is_open()) {
64         perror("Error opening file");
65         exit(1);
66     }
67
68     cout << "The contents of the file " << filename << ":\n";
69     string line;
70     while (getline(file, line)) {
71         cout << line << "\n";
72     }
73     cout << endl;
74
75     file.close();
76 }

```

```

77
78 int getLastLineWordCount(const char *filename) {
79     ifstream file(filename);
80     if (!file.is_open()) {
81         perror("Error opening file");
82         exit(1);
83     }
84
85     string line, lastLine;
86     while (getline(file, line)) {
87         lastLine = line;
88     }
89     file.close();
90
91     return countWords(lastLine);
92 }
93
94 int main() {
95     const char *file1 = "F1.txt";
96     const char *file2 = "F2.txt";
97     int N1 = 3, N2 = 5;
98
99     createFileF1(file1);
100    printf("\n");
101    printFile(file1);
102    printf("\n");
103    copyToFileF2(file1, file2, N1, N2);
104
105    printFile(file2);
106    printf("\n");
107    int wordCount = getLastLineWordCount(file2);
108    cout << "The number of words in the last line of the file F2: " << wordCount << "\n";
109
110    return 0;
111 }

```

John Black - Epic 5 Task 6 - Lab# programming: Algotester Lab 4(варіант 2 з використанням STL)

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5
6  int main() {
7      int N, K;
8      cin >> N >> K;
9
10     vector<int> a(N);
11     for (int i = 0; i < N; ++i) {
12         cin >> a[i];
13     }
14
15     sort(a.begin(), a.end());
16     auto last = unique(a.begin(), a.end());
17     a.erase(last, a.end());
18
19     K %= a.size();
20     rotate(a.begin(), a.begin() + K, a.end());
21
22     cout << a.size() << endl;
23     for (int x : a) {
24         cout << x << " ";
25     }
26     cout << endl;
27
28     return 0;
29 }

```

John Black - Epic 5 Task 6 - Lab# programming: Algotester Lab 4(варіант 2 з власною імплементацією)

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5
6  void sort(vector<int>& arr) {
7      int n = arr.size();
8      for (int i = 0; i < n - 1; i++) {
9          int minIndex = i;
10         for (int j = i + 1; j < n; j++) {
11             if (arr[j] < arr[minIndex]) {
12                 minIndex = j;
13             }
14         }
15         if (minIndex != i) {
16             swap(arr[i], arr[minIndex]);
17         }
18     }
19 }
20
21 vector<int> unique(vector<int>& arr) {
22     vector<int> result;
23     for (int i = 0; i < arr.size(); i++) {
24         if (result.empty() || arr[i] != result.back()) {
25             result.push_back(arr[i]);
26         }
27     }
28     return result;
29 }
30
31 vector<int> reverse(vector<int>& arr, int K) {
32     int t = arr.size();
33     K %= t;
34     vector<int> result(t);
35
36     for (int i = 0; i < t; i++) {
37         result[(i + t - K) % t] = arr[i];
38     }
39
40     return result;
41 }
42
43 int main() {
44     int N, K;
45     cin >> N >> K;
46
47     vector<int> a(N);
48     for (int i = 0; i < N; i++) {
49         cin >> a[i];
50     }
51
52     sort(a);
53     vector<int> filtered = unique(a);
54     vector<int> reversed = reverse(filtered, K);
55
56     cout << filtered.size() << endl;
57
58     for (int i = 0; i < reversed.size(); i++) {
59         cout << reversed[i] << " ";
60     }
61     cout << endl;
62
63     return 0;
64 }
```

John Black - Epic 5 Task 7 - Lab# programming: Algotester Lab 6(вариант 3)

```
1  #include <iostream>
2  #include <vector>
3  #include <set>
4  #include <string>
5
6  using namespace std;
7
8  int main() {
9      int N;
10     cin >> N;
11
12     // Зчитуємо поле
13     vector<vector<int>> field(N, vector<int>(N));
14     for (int i = 0; i < N; i++) {
15         string row;
16         cin >> row;
17         for (int j = 0; j < N; j++) {
18             field[i][j] = row[j] - '0';
19         }
20     }
21
22     int Q;
23     cin >> Q;
24
25     vector<string> results;
26
27     for (int t = 0; t < Q; t++) {
28         int x, y;
29         cin >> x >> y;
30         x = x - 1;
31         y = y - 1;
32
33         set<int> cantused;
34
35
36         for (int i = 0; i < N; i++) {
37             if (field[x][i] != 0) {
38                 cantused.insert(field[x][i]);
39             }

```



```

40     }
41
42     for (int i = 0; i < N; i++) {
43         if (field[i][y] != 0) {
44             cantused.insert(field[i][y]);
45         }
46     }
47
48
49     if (field[x][y] != 0) {
50         string result = "1 " + to_string(field[x][y]);
51         results.push_back(result);
52     } else {
53
54         vector<int> possible;
55         for (int num = 1; num <= N; num++) {
56             if (cantused.find(num) == cantused.end()) {
57                 possible.push_back(num);
58             }
59         }
60
61
62         string result = to_string(possible.size());
63         for (int num : possible) {
64             result += " " + to_string(num);
65         }
66         results.push_back(result);
67     }
68 }
69
70 cout << endl;
71
72 for (const string &res : results) {
73     cout << res << "\n";
74     cout << endl;
75 }
76
77 return 0;
78 }

```

John Black - Epic 5 Task 8 - Practice# programming: Class Practice Task(завдання 1)

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  using namespace std;
6
7  enum class FileOpResult {
8      Success,
9      Failure
10 };
11
12 FileOpResult write_to_file(const char *name, const string content[], int quantity_of_lines) {
13     if (name == nullptr || quantity_of_lines <= 0) {
14         cerr << "Error: Invalid arguments provided.\n";
15         return FileOpResult::Failure;
16     }
17
18     ofstream file(name, ios::out | ios::trunc);
19     if (!file) {
20         cerr << "Error: Unable to open file for writing.\n";
21         return FileOpResult::Failure;
22     }
23
24     for(int i = 0; i < quantity_of_lines; i++){
25         file << content[i] << endl;
26
27         if (!file) {
28             cerr << "Error: Write operation failed on line " << i + 1 << ".\n";
29             file.close();
30             return FileOpResult::Failure;
31         }
32     }
33
34     file.close();
35     if (!file) {
36         cerr << "Error: Failed to close the file.\n";
37         return FileOpResult::Failure;
38     }
39
40     return FileOpResult::Success;
41 }
42
43 int main() {
44     int quantity_of_lines;
45     char filename[256];
46
47     cout << "Enter file name: ";
48     cin.getline(filename, sizeof(filename));
49
50     cout << "Enter number of lines: ";
51     cin >> quantity_of_lines;
52     cin.ignore();
53
54
55     string content[quantity_of_lines];
56
57     cout << "Enter content to write into the file: ";
58     for(int i = 0; i < quantity_of_lines; i++){
59         getline(cin, content[i]);
60     }
61     FileOpResult result = write_to_file(filename, content, quantity_of_lines);
62
63     if (result == FileOpResult::Success) {
64         cout << "File written successfully.\n";
65     } else {
66         cerr << "Failed to write to the file.\n";
67     }
68
69     return 0;
70 }

```

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  using namespace std;
6
7  // C:\Users\Yura\Desktop\c++\epic_5(1).txt шлях до файлів
8  // C:\Users\Yura\Desktop\c++\epic_5(2).txt
9
10 enum class FileOpResult {
11     Success,
12     Failure
13 };
14
15 FileOpResult copy_file(char *file_from, char *file_to) {
16     if (file_from == nullptr || file_to == nullptr) {
17         cerr << "Error: Invalid files provided.\n";
18         return FileOpResult::Failure;
19     }
20
21     ifstream from(file_from, ios::in | ios::binary);
22     if (!from) {
23         cerr << "Error: Unable to open file for copying.\n";
24         return FileOpResult::Failure;
25     }
26
27     ofstream to(file_to, ios::out | ios::binary);
28     if (!to) {
29         cerr << "Error: Unable to open file for pasting.\n";
30         return FileOpResult::Failure;
31     }
32
33     to << from.rdbuf();
34
35     if (!from || !to) {
36         cerr << "Error: File copy operation failed.\n";
37         return FileOpResult::Failure;
38     }
39     from.close();
40     to.close();
41
42     return FileOpResult::Success;
43 }
44
45 int main() {
46     char file_from[256];
47     char file_to[256];
48
49     cout << "Enter the name of the source file: ";
50     cin.getline(file_from, sizeof(file_from));
51
52     cout << "Enter the name of the destination file: ";
53     cin.getline(file_to, sizeof(file_to));
54
55     if (copy_file(file_from, file_to) == FileOpResult::Success) {
56         cout << "File copied successfully.\n";
57     } else {
58         cerr << "File copy failed.\n";
59     }
60
61     return 0;
62 }

```

John Black - Epic 5 Task 9 - Practice# programming: Self Practice Task

```
1  #include <iostream>
2  #include <fstream>
3  #include <sstream>
4  #include <string>
5  #include <cctype>
6  #include <vector>
7  #include <map>
8
9  using namespace std;
10
11  // Files:
12  // input.txt
13  // output.txt
14  // capitalized_words.txt
15
16  int main() {
17      string input, output, capitalized_words;
18
19      cout << "Enter name of input file: ";
20      getline(cin, input);
21      cout << "Enter name of output file: ";
22      getline(cin, output);
23      cout << "Enter name of capitalized words file: ";
24      getline(cin, capitalized_words);
25
26      ifstream inputFile(input);
27      ofstream outputFile(output);
28      ofstream capitalizedWordsFile(capitalized_words);
29
30      if (!inputFile.is_open() || !outputFile.is_open() || !capitalizedWordsFile.is_open()) {
31          cerr << "Error opening files!" << endl;
32          return 1;
33      }
34
35      string line;
36      int lineCount = 0, wordCount = 0, charCount = 0;
37      vector<string> capitalizedWords;
38
39      while (getline(inputFile, line)) {
```

```

40     lineCount++;
41     charCount += line.size() + 1;
42     string word = "";
43
44     for (char ch : line) {
45         if (isalnum(ch)) {
46             word += ch;
47         } else {
48             if (!word.empty()) {
49                 wordCount++;
50
51                 if (isupper(word[0])) {
52                     capitalizedWords.push_back(word);
53                 }
54                 word = "";
55             }
56         }
57     }
58
59     if (!word.empty()) {
60         wordCount++;
61         if (isupper(word[0])) {
62             capitalizedWords.push_back(word);
63         }
64     }
65 }
66
67 outputFile << "Lines: " << lineCount << endl;
68 outputFile << "Words: " << wordCount << endl;
69 outputFile << "Chars: " << charCount << endl;
70
71 for (const string &word : capitalizedWords) {
72     capitalizedWordsFile << word << endl;
73 }
74
75 inputFile.close();
76 outputFile.close();
77
78 capitalizedWordsFile.close();
79
79 cout << "Everything is written." << endl;
80 return 0;
81 }

```

5)Результати виконання завдань та фактично затрачений час

John Black - Epic 5 Task 3 - Lab# programming: VNS Lab 6(варіант 17)

```
Enter line:  
hi this is testing of program  
Sorted words:  
hi  
is  
of  
program  
testing  
this
```

Фактичний час виконання: 28хв

John Black - Epic 5 Task 4 - Lab# programming: VNS Lab 8(варіант 17)

Contents of the file after changes:

Name: The Godfather

Director: Francis Ford Coppola

Year of release: 1972

Price: 9.99

Name: The Shawshank Redemption

Director: Frank Darabont

Year of release: 1994

Price: 14.99

Name: Inception

Director: Christopher Nolan

Year of release: 2010

Price: 15.99

Name: Memento

Director: Christopher Nolan

Year of release: 2000

Price: 12.99

Name: The Dark Knight

Director: Christopher Nolan

Year of release: 2008

Price: 19.99

The contents of the file after deleting the element:

Name: The Godfather

Director: Francis Ford Coppola

Year of release: 1972

Price: 9.99

Name: Inception

Director: Christopher Nolan

Year of release: 2010

Price: 15.99

Name: Memento

Director: Christopher Nolan

Year of release: 2000

Price: 12.99

Name: The Dark Knight

Director: Christopher Nolan

Year of release: 2008

Price: 19.99

Фактичний час виконання: 2 год 14хв

John Black - Epic 5 Task 5 - Lab# programming: VNS Lab 9(варіант 4)


The contents of the file F1.txt:

Autumn leaves fell gracefully from the trees, creating a colorful carpet on the ground.
The old, wooden house stood at the end of the street, its windows dark and mysterious.
A gentle breeze carried the sweet scent of roses through the garden.
Children laughed and played on the beach as the waves crashed against the shore.
The stars twinkled brightly in the night sky, offering a breathtaking view.
Anxious about the upcoming exam, she tried to focus on her studies.
The cat curled up on the soft blanket, purring contentedly.
He enjoyed hiking in the mountains, breathing the fresh air.
The book was so captivating, she couldn't put it down.
After a long day at work, she was looking forward to relaxing at home.

The contents of the file F2.txt:


A gentle breeze carried the sweet scent of roses through the garden.
He enjoyed hiking in the mountains, breathing the fresh air.
The book was so captivating, she couldn't put it down.
After a long day at work, she was looking forward to relaxing at home.

The number of words in the last line of the file F2: 14

 F1.txt: Блокнот

Файл Редагування Формат Вигляд Довідка

Autumn leaves fell gracefully from the trees, creating a colorful carpet on the ground.
The old, wooden house stood at the end of the street, its windows dark and mysterious.
A gentle breeze carried the sweet scent of roses through the garden.
Children laughed and played on the beach as the waves crashed against the shore.
The stars twinkled brightly in the night sky, offering a breathtaking view.
Anxious about the upcoming exam, she tried to focus on her studies.
The cat curled up on the soft blanket, purring contentedly.
He enjoyed hiking in the mountains, breathing the fresh air.
The book was so captivating, she couldn't put it down.
After a long day at work, she was looking forward to relaxing at home.

 F2.txt: Блокнот

Файл Редагування Формат Вигляд Довідка

A gentle breeze carried the sweet scent of roses through the garden.
He enjoyed hiking in the mountains, breathing the fresh air.
The book was so captivating, she couldn't put it down.
After a long day at work, she was looking forward to relaxing at home.

Фактичний час виконання: 58хв

John Black - Epic 5 Task 6 - Lab# programming: Algotester Lab 4(варіант 2 з використанням STL)

```
10
11
5 6 2 3 1 2 3 3 4 7
7
5 6 7 1 2 3 4
```

Фактичний час виконання: 1год 12хв

John Black - Epic 5 Task 6 - Lab# programming: Algotester Lab 4(варіант 2 з власною імплементацією)

```
10
11
5 6 2 3 1 2 3 3 4 7
7
5 6 7 1 2 3 4
PS C:\Users\Yura\Desktop\c++> ^C
PS C:\Users\Yura\Desktop\c++>
PS C:\Users\Yura\Desktop\c++> & 'c:\
ncher.exe' '--stdin=Microsoft-MIEngin
r-4wheqabi.3qi' '--pid=Microsoft-MIEn
10
3
1 2 2 3 3 3 4 5 6 7
7
4 5 6 7 1 2 3
```

Фактичний час виконання: 2год 25хв

John Black - Epic 5 Task 7 - Lab# programming: Algotester Lab 6(варіант 3)

```
9
123456789
000000000
010000000
030000000
040000000
050000000
060000000
070000000
987643215
3
1 1
2 2
8 8

1 1

1 9

6 2 3 4 5 6 9
```

Фактичний час виконання: 1год

John Black - Epic 5 Task 8 - Practice# programming: Class Practice Task(завдання 1)

```
Enter file name: epic_5(1).txt
Enter number of lines: 1
Enter content to write into the file: Hi me, from future. I hope you have successfully pass 5th epic.
File written successfully.
PS C:\Users\Yura\Desktop\c++>
```



epic_5(1).txt: Блокнот


Файл Редагування Формат Вигляд Довідка

Hi me, from future. I hope you have successfully pass 5th epic.

Фактичний час виконання: 32хв


John Black - Epic 5 Task 8 - Practice# programming: Class Practice Task(завдання 2)

```
Enter the name of the source file: epic_5(1).txt
Enter the name of the destination file: epic_5(2).txt
File copied successfully.
PS C:\Users\Yura\Desktop\c++>
```

 epic_5(1).txt: Блокнот

Файл Редагування Формат Вигляд Довідка

Hi me, from future. I hope you have successfully pass 5th epic.

 epic_5(2).txt: Блокнот


Файл Редагування Формат Вигляд Довідка

Hi me, from future. I hope you have successfully pass 5th epic.

Фактичний час виконання: 23хв

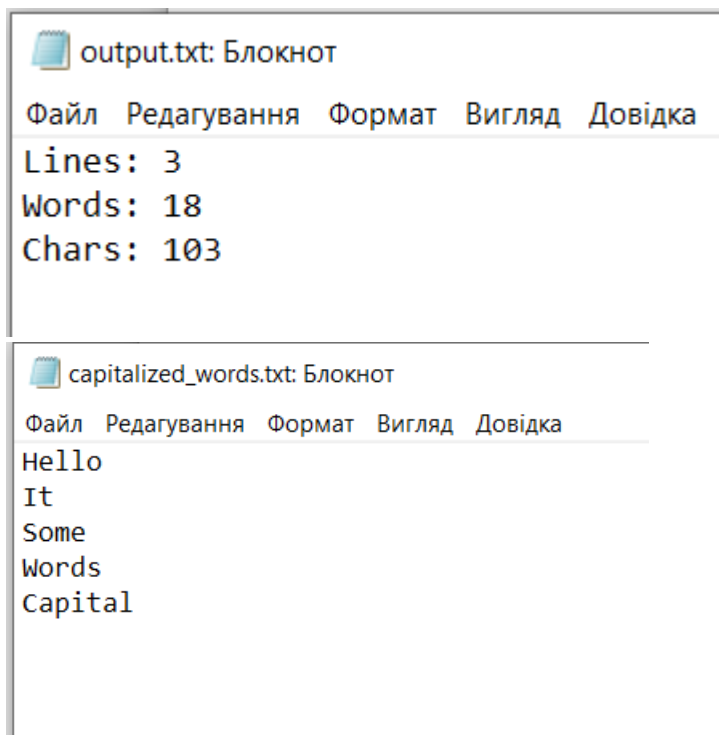
John Black - Epic 5 Task 9 - Practice# programming: Self Practice Task

```
Enter name of input file: input.txt
Enter name of output file: output.txt
Enter name of capitalized words file: capitalized_words.txt
Everything is written.
PS C:\Users\Yura\Desktop\c++>
```

 input.TXT: Блокнот

Файл Редагування Формат Вигляд Довідка

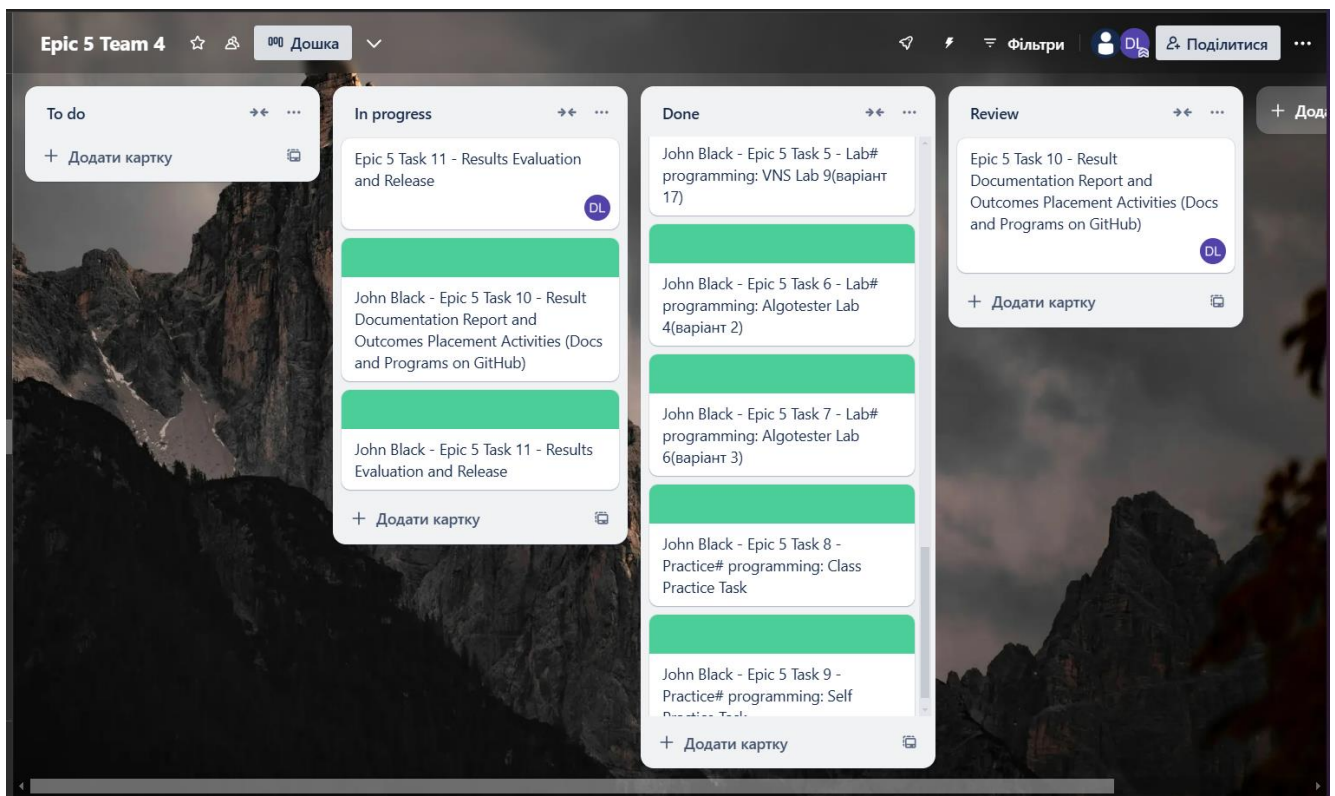
Hello, this is a sample text file.
It contains multiple lines.
Some Words start with a Capital letter.



Фактичний час виконання: 1год 29хв

б)Робота з комадою

Trello:



Висновок: Робота з файлами, як текстовими, так і бінарними, є важливим елементом програмування, що дозволяє зберігати, обробляти та передавати дані. Стандартна бібліотека надає зручний набір інструментів для роботи з файлами, які спрощують операції читання, запису та обробки даних. Володіння цими навичками є важливим для створення ефективного та масштабованого програмного забезпечення, а також для розробки власних бібліотек, що адаптовані до специфічних потреб.

Посилання на пул реквест: