

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(ла):

Студент групи ІІІ-12
Соснило Богдан Тарасович

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

Мета: Опанувати роботу з файлами у мові C++: навчитися створювати, відкривати, читати та записувати дані, працювати з текстовими і бінарними файлами. Розвинути практичні навички використання рядкових змінних під час обробки даних. Ознайомитися зі Стандартною бібліотекою C++, зрозуміти її принципи та навчитися ефективно застосовувати класи та методи для роботи з файлами.

Теоретичні відомості:

1) Файли:

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

<http://cpp.dp.ua/ponyattya-fajla/>

<http://cpp.dp.ua/vykorystannya-fajliv/>

Опрацювання теорії по файлах допомогло мені написати програми, де треба було записувати інформацію у файл, читати інформацію з файла, оперувати текстовими файлами і бінарними файлами.

2) Рядки:

<https://acode.com.ua/urok-60-vvedennya-v-std-string/>

<https://acode.com.ua/urok-82-ryadky-c-style/>

<http://cpp.dp.ua/vykorystannya-ryadkiv-typu-string/>

Ознайомлення з рядками дозволило мені вміти їх задавати різними способами, оперувати ними, знаходити слова у рядках, посилатись до рядків через вказівники, що я і використовував у зроблених мною задачах.

3) Стандартна бібліотека:

<https://acode.com.ua/urok-204-standartna-biblioteka-shabloniv-stl/>

<https://acode.com.ua/urok-205-kontejnery-stl/>

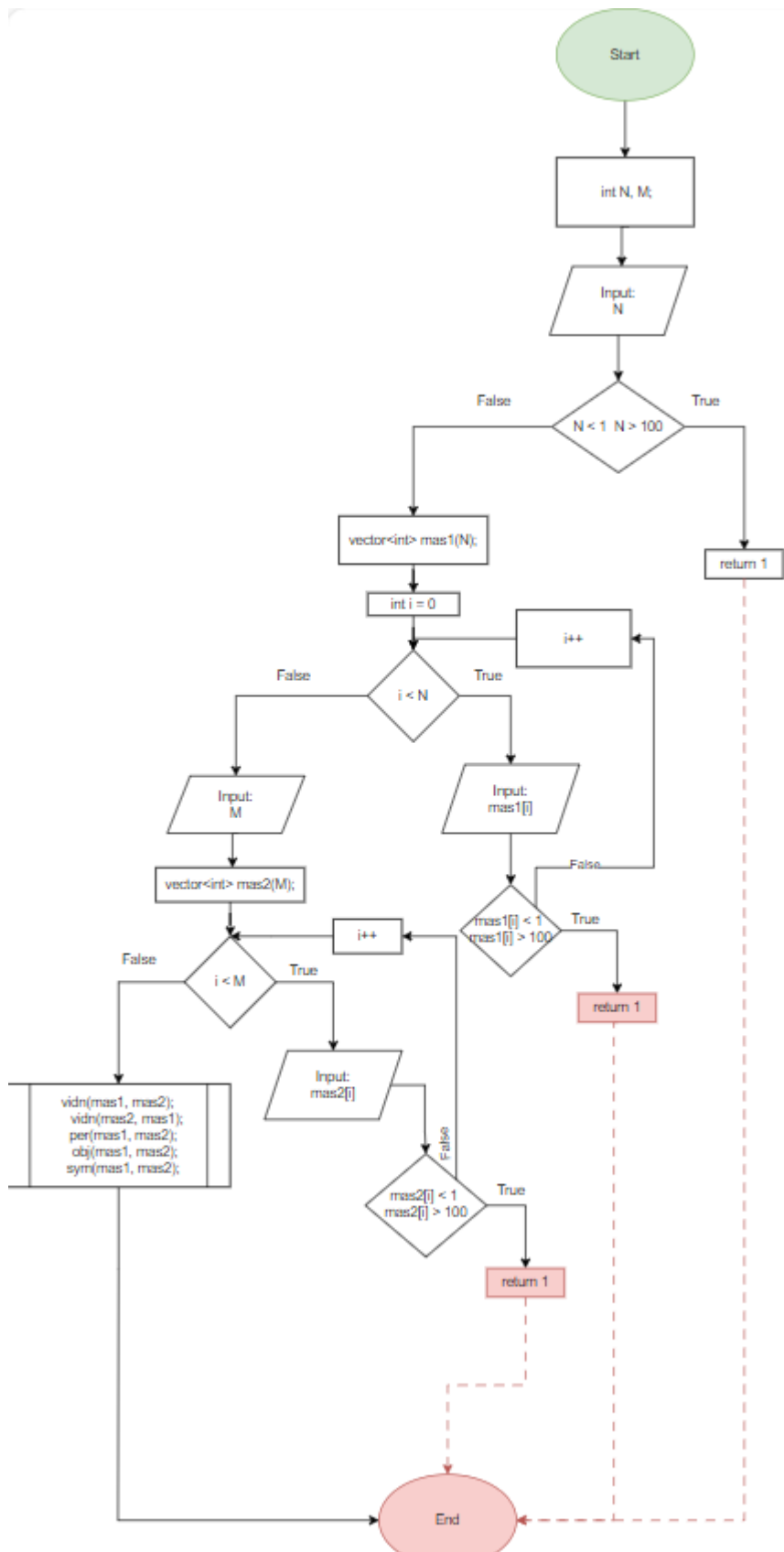
<https://acode.com.ua/urok-206-iteratory-stl/>

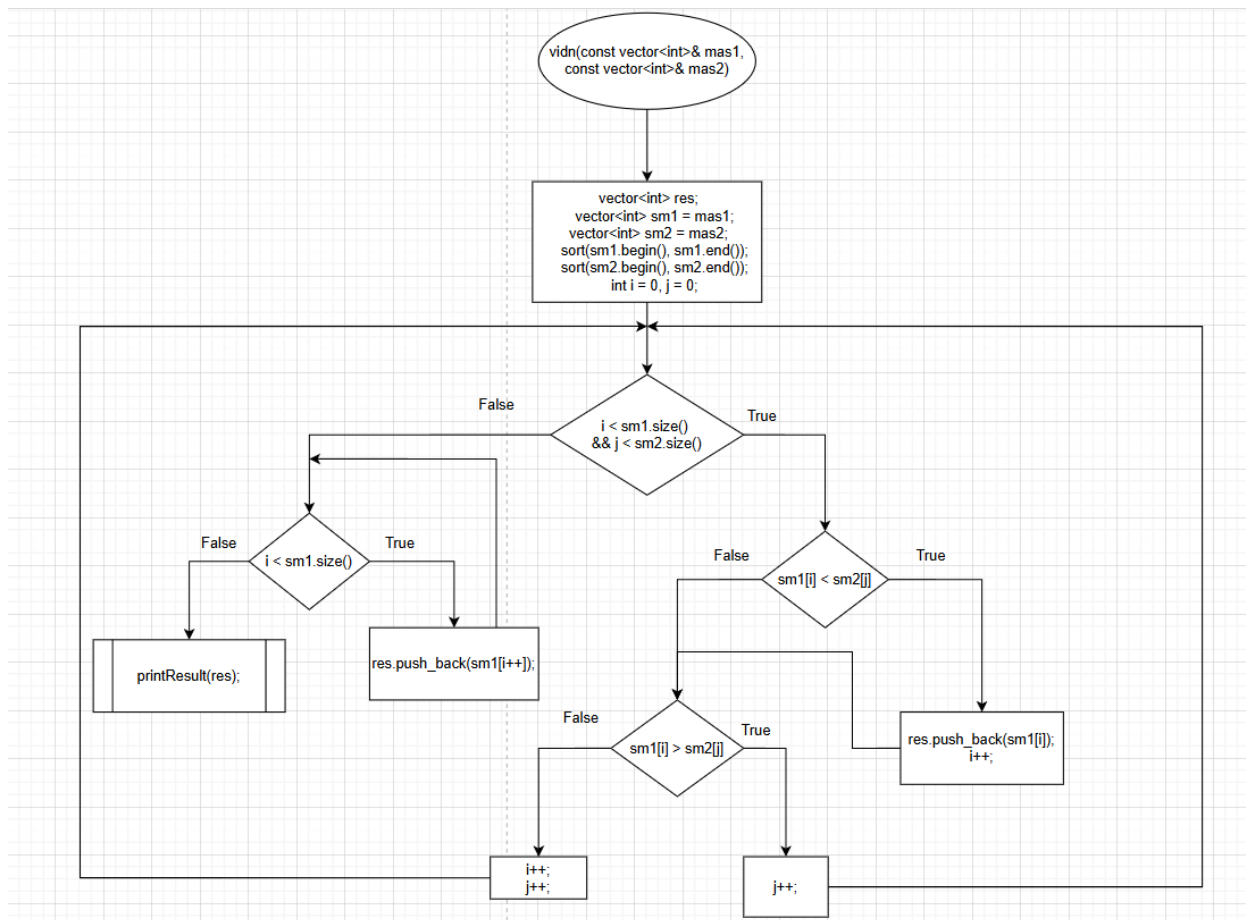
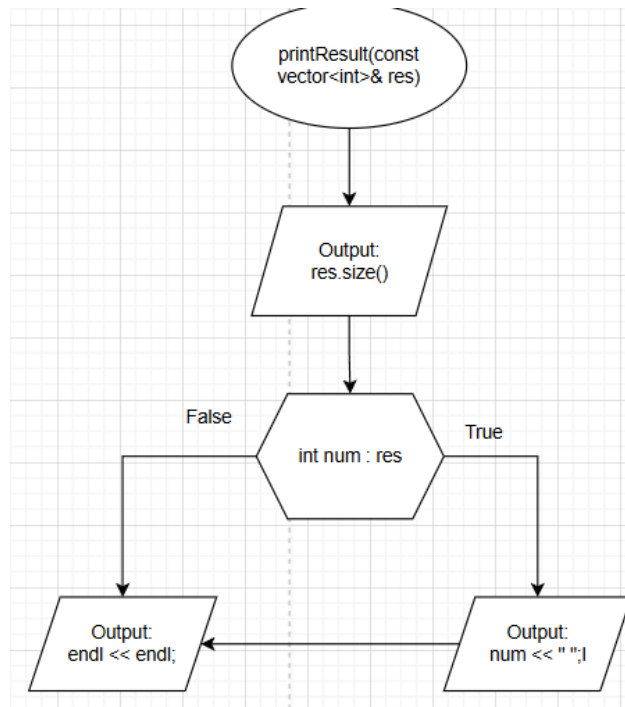
<https://acode.com.ua/urok-207-algorytmy-stl/>

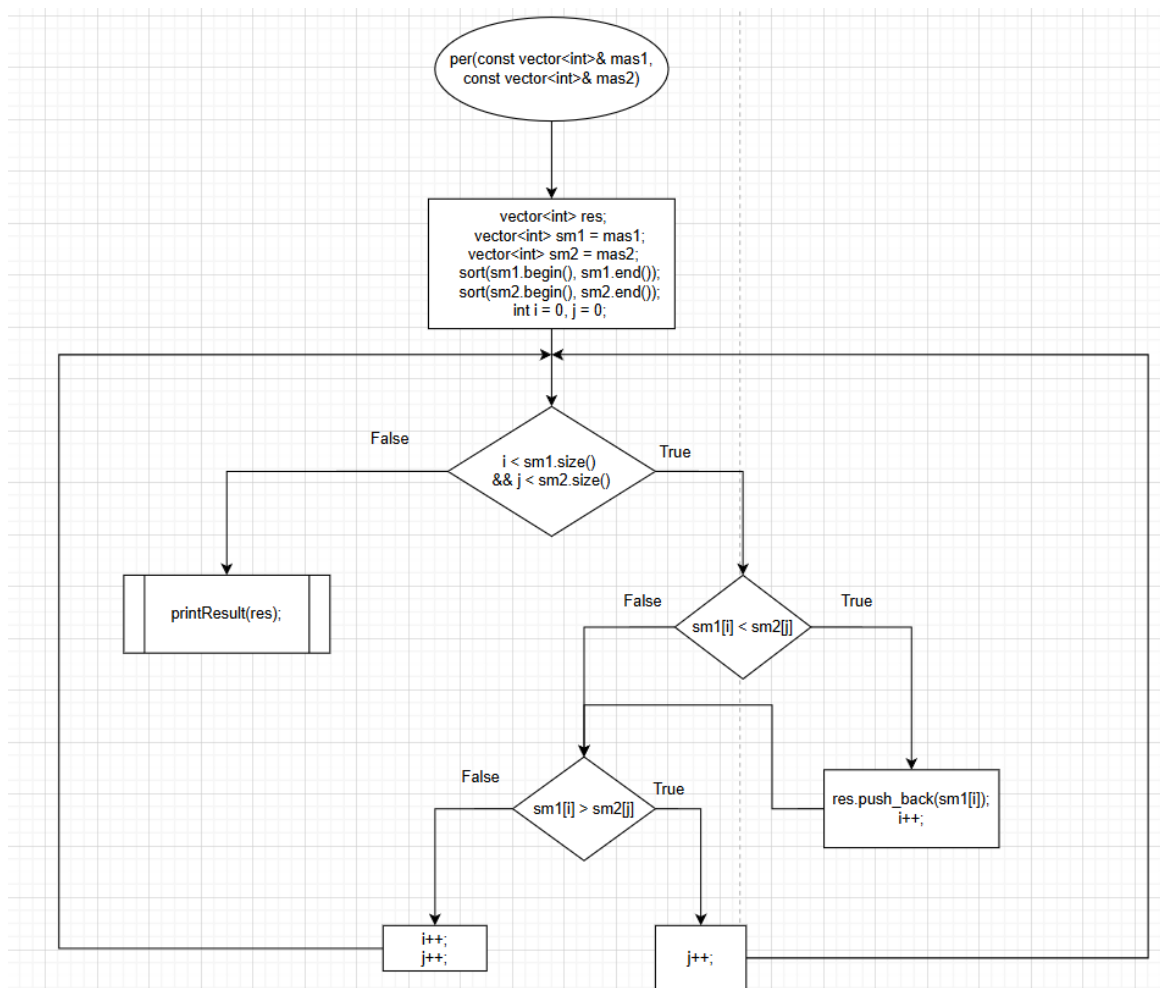
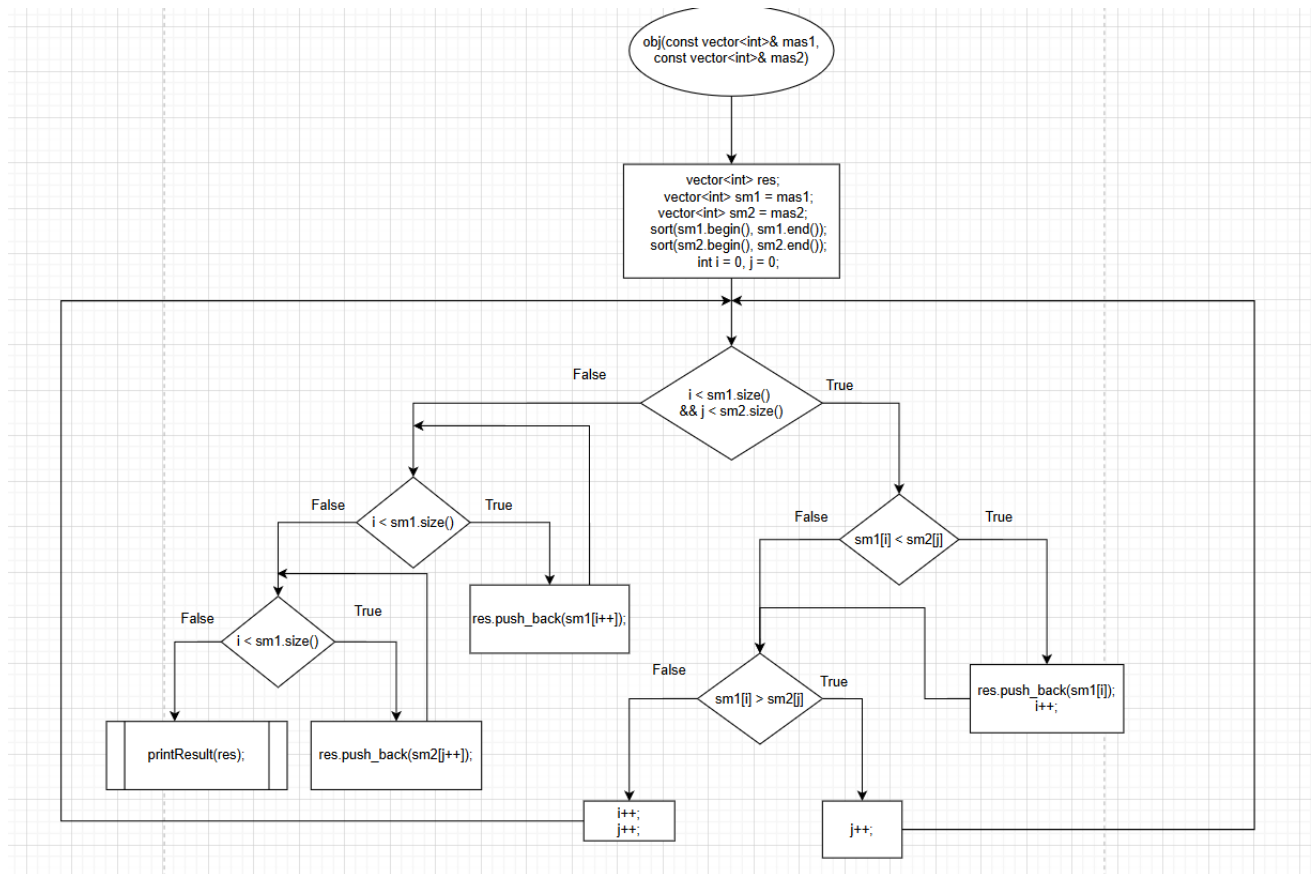
Ознайомившись з стандартними бібліотеками, я зрозумів, що це дуже корисна річ, якою можна полегшити роботу з кодом.

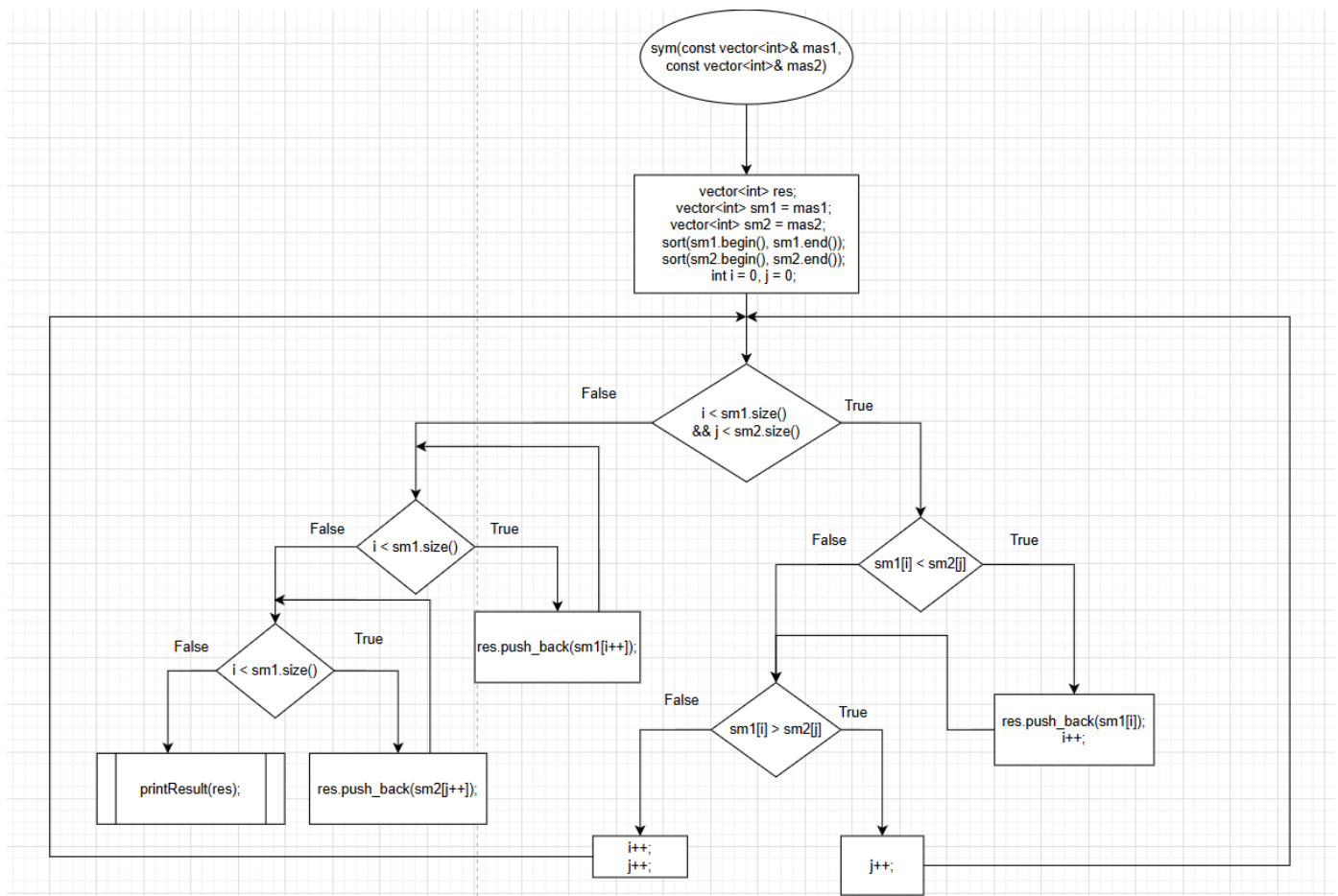
Виконання роботи

Task 2 - Requirements management (understand tasks) and design activities (draw flow diagrams and estimate tasks 3-7) (2 год)









Task 3 - Lab# programming: VNS Lab 6 (зайняло в мене 30 хв)

Задано рядок, що складається із символів. Надрукувати всі слова, які співпадають з його першим словом.

```

1  #include <iostream>
2  #include <stdio.h>
3  #include <string.h>
4
5  using namespace std;
6
7  int funct(char str[], char str1[]) {
8      char* pos = str;
9      int i = 0;
10     while ((pos = strstr(pos, str1)) != nullptr) {
11         i++;
12         pos += strlen(str1);
13     }
14     return i;
15 }
16
17 int main() {
18     int i = 0;
19     char str1[256];
20     char str[256];
21
22     cout << "Введіть рядок: ";
23     gets(str);
24
25     while (str[i] != ' ' && str[i] != '\0') {
26         str1[i] = str[i];
27         i++;
28     }
29     str1[i] = '\0';
30
31     for (i = 0; i < funct(str, str1); i++) {
32         cout << str1 << " ";
33     }
34     return 0;
35 }
36

```

Результат:

```

Введіть рядок: opa rrrr ooooooff asads opa opa
opa opa opa

```

Task 4 - Lab# programming: VNS Lab 8 (1.5 год)

Структура "Людина":

- прізвище, ім'я, по батькові;
- домашня адреса;
- номер телефону;
- вік.

Знищити усі елементи із заданим віком, додати елемент після елемента із заданим номером.

```
1  #include <iostream>
2  #include <fstream>
3  #include <vector>
4  #include <string>
5  #include <locale>
6
7  using namespace std;
8
9  struct Human {
10     char surname[50];
11     char name[50];
12     char patronymic[50];
13     char address[100];
14     char phone[20];
15     int age;
16 };
17
18 void inputHumans(vector<Human>& humans) {
19     int count;
20     cout << "Введіть кількість людей: ";
21     cin >> count;
22     cin.ignore();
23
24     for (int i = 0; i < count; ++i) {
25         Human human;
26         cout << "\nВведіть дані про людину " << (i + 1) << ": " << endl;
27
28         cout << "Прізвище: ";
29         cin.getline(human.surname, 50);
30         cout << "Ім'я: ";
31         cin.getline(human.name, 50);
32         cout << "По-батькові: ";
33         cin.getline(human.patronymic, 50);
34         cout << "Адреса: ";
35         cin.getline(human.address, 100);
36         cout << "Телефон: ";
37         cin.getline(human.phone, 20);
38         cout << "Вік: ";
39         cin >> human.age;
40         cin.ignore();
41         humans.push_back(human);
42     }
43 }
44
45 void createBinaryFile(const string& filename, const vector<Human>& humans) {
46     ofstream file(filename, ios::binary);
47     if (!file) {
48         cerr << "Помилка відкриття файлу!" << endl;
49         return;
50     }
51     for (const auto& human : humans) {
52         file.write(reinterpret_cast<const char*>(&human), sizeof(Human));
53     }
54     file.close();
55 }
```



```

55 }
56
57
58 void printBinaryFile(const string& filename) {
59     ifstream file(filename, ios::binary);
60     if (!file) {
61         cerr << "Помилка відкриття файлу!" << endl;
62         return;
63     }
64
65     Human human;
66     while (file.read(reinterpret_cast<char*>(&human), sizeof(Human))) {
67         cout << human.surname << " " << human.name << " " << human.patronymic
68             << ", " << human.address << ", " << human.phone
69             << ", вік: " << human.age << endl;
70     }
71
72     file.close();
73 }
74
75
76 void deleteByAge(const string& filename, int ageToDelete) {
77     ifstream inFile(filename, ios::binary);
78     ofstream outFile("temp.dat", ios::binary);
79     if (!inFile || !outFile) {
80         cerr << "Помилка відкриття файлу!" << endl;
81         return;
82     }
83
84     Human human;
85     while (inFile.read(reinterpret_cast<char*>(&human), sizeof(Human))) {
86         if (human.age != ageToDelete) {
87             outFile.write(reinterpret_cast<const char*>(&human), sizeof(Human));
88         }
89     }
90
91     inFile.close();
92     outFile.close();
93     remove(filename.c_str());
94     rename("temp.dat", filename.c_str());
95 }
96
97 // Функція для додавання запису після елемента з номером телефону
98 void addAfterPhone(const string& filename, const string& phone, const Human& newHuman) {
99     ifstream inFile(filename, ios::binary);
100     ofstream outFile("temp.dat", ios::binary);
101     if (!inFile || !outFile) {
102         cerr << "Помилка відкриття файлу!" << endl;
103         return;
104     }
105
106     Human human;
107     while (inFile.read(reinterpret_cast<char*>(&human), sizeof(Human))) {
108         outFile.write(reinterpret_cast<const char*>(&human), sizeof(Human));
109         if (phone == human.phone) {

```

```

110             outFile.write(reinterpret_cast<const char*>(&newHuman), sizeof(Human));
111         }
112     }
113
114     inFile.close();
115     outFile.close();
116     remove(filename.c_str());
117     rename("temp.dat", filename.c_str());
118 }
119
120 int main() {
121
122     setlocale(LC_ALL, "uk_UA.UTF-8");
123
124     string filename = "humans.dat";
125     vector<Human> humans;
126
127     inputHumans(humans);
128     createBinaryFile(filename, humans);
129
130     cout << "\nПоточковий вміст файлу:" << endl;
131     printBinaryFile(filename);

```

```

131     printBinaryFile(filename);
132
133     int ageToDelete;
134     cout << "\nВведіть вік для видалення записів: ";
135     cin >> ageToDelete;
136     deleteByAge(filename, ageToDelete);
137
138     cout << "\nПісля видалення записів із віком " << ageToDelete << ":" << endl;
139     printBinaryFile(filename);
140
141     Human newHuman;
142     cout << "\nВведіть дані нового запису для додавання:\n";
143     cin.ignore();
144     cout << "Прізвище: "; cin.getline(newHuman.surname, 50);
145     cout << "Ім'я: "; cin.getline(newHuman.name, 50);
146     cout << "По-батькові: "; cin.getline(newHuman.patronymic, 50);
147     cout << "Адреса: "; cin.getline(newHuman.address, 100);
148     cout << "Телефон: "; cin.getline(newHuman.phone, 20);
149     cout << "Вік: "; cin >> newHuman.age;
150
151     string phoneToFind;
152     cout << "\nВведіть номер телефону, після якого додати новий запис: ";
153     cin >> phoneToFind;
154     addAfterPhone(filename, phoneToFind, newHuman);
155
156     cout << "\nПісля додавання нового запису телефону " << phoneToFind << ":" << endl;
157     printBinaryFile(filename);
158
159     return 0;
160 }

```

```

Введіть кількість людей: 2

Введіть дані про людину 1:
Прізвище: Oliiuk
Ім'я: Bozhena
По-батькові: Orestivna
Адреса: naukova
Телефон: +380949344930
Вік: 18

Введіть дані про людину 2:
Прізвище: Tokaryl
Ім'я: Serhii
По-батькові: Vladislavovych
Адреса: saharova
Телефон: +380847387434
Вік: 19

Початковий вміст файлу:
Oliiuk Bozhena Orestivna, naukova, +380949344930, вік: 18
Tokaryl Serhii Vladislavovych, saharova, +380847387434, вік: 19

Введіть вік для видалення записів: 19

Після видалення записів із віком 19:
Oliiuk Bozhena Orestivna, naukova, +380949344930, вік: 18

Введіть дані нового запису для додавання:
Прізвище: sosnylo
Ім'я: bohdan
По-батькові: andrievich
Адреса: chuprinki
Телефон: +30934040404
Вік: 20

Введіть номер телефону, після якого додати новий запис: +380949344930

Після додавання нового запису телефону +380949344930:
Oliiuk Bozhena Orestivna, naukova, +380949344930, вік: 18
sosnylo bohdan andrievich, chuprinki, +30934040404, вік: 20
PS C:\Users\User\Desktop\C++\epic_5>

```

Task 5 - Lab# programming: VNS Lab 9 (30 хвилин)

Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього Інформацію. 1) Скопіювати з файлу F1 у файл F2 рядки, починаючи з 4.

2) Підрахувати кількість символів в останньому слові F2.

```
1  #include <iostream>
2  #include <fstream>
3  using namespace std;
4
5  void formfile(const string& filename){
6      ofstream file(filename);
7      if (!file) {
8          cerr << "Помилка відкриття файлу!" << endl;
9          return;
10     }
11     cout << "к-сть рядків(не менше 10): ";
12     int k;
13     cin >> k;
14
15     for(int i = 0; i<k;i++){
16         string m;
17         cin >> m;
18         file << m << " ";
19     }
20
21     file.close();
22 }
23
24 void inputinf(const string& filename, const string& filename1){
25     ifstream file(filename);
26     ofstream file1(filename1);
27     if(!file || !file1){
28         cerr << "Помилка відкриття файлу!" << endl;
29         return;
30     }
31     int i = 0;
32     string strinput;
33     while(file >> strinput){
34         i++;
35         if(i >=4){
36             file1 << strinput << " ";
37         }
38     }
39
40     file1.close();
41     file.close();
42     ifstream file2(filename1);
43     string strinput1;
44     string last1;
45     while(file2 >> strinput1){
46         last1.assign(strinput1);
47     }
48
49     if (!last1.empty()) {
50         cout << "Кількість символів в останньому слові файлу F2: " << last1.length() << endl;
51     } else {
52         cout << "Файл F2 порожній." << endl;
53     }
54 }
```

```
53     }
54
55 }
56
57 int main(){
58
59     string file = "F1.txt";
60     string file1 = "F2.txt";
61     formfile(file);
62     inputinf(file, file1);
63
64     return 0;
65 }
```

```
ALALLA afafafa rarkarkar hfdsjfdsjf hjhbdjbhjbhd
Кількість символів в останньому слові файлу F2: 12
```

```
epic_5 > F1.txt
1 ALALLA afafafa rarkarkar hfdsjfdsjf hjhbdjbhjbhd
```

```
epic_5 > F2.txt
1 hfdsjfdsjf hjhbdjbhjbhd
```

Task 6 - Lab# programming: Algotester Lab 4

A) з stl:

Вам дано 2 цілих чисел масиви, розміром N та M .

Ваше завдання вивести:

1. Різницю $N-M$
2. Різницю $M-N$
3. Їх перетин
4. Їх об'єднання
5. Їх симетричну різницю

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 #include <iterator>
5
6 using namespace std;
7
8 void printResult(const vector<int>& res) {
9     cout << res.size() << endl;
10    for (int i : res) {
11        cout << i << " ";
12    }
13    cout << endl << endl;
14 }
15
16 void vidn(vector<int>& mas1, vector<int>& mas2) {
17     vector<int> res;
18     sort(mas1.begin(), mas1.end());
19     sort(mas2.begin(), mas2.end());
20
21     set_difference(mas1.begin(), mas1.end(), mas2.begin(), mas2.end(), back_inserter(res));
22     printResult(res);
23 }
24
25 void per(vector<int>& mas1, vector<int>& mas2) {
26     vector<int> res;
27     sort(mas1.begin(), mas1.end());
28     sort(mas2.begin(), mas2.end());
29
30     set_intersection(mas1.begin(), mas1.end(), mas2.begin(), mas2.end(), back_inserter(res));
31     printResult(res);
32 }
33
34 void obj(vector<int>& mas1, vector<int>& mas2) {
35     vector<int> res;
36     sort(mas1.begin(), mas1.end());
37     sort(mas2.begin(), mas2.end());
38
39     set_union(mas1.begin(), mas1.end(), mas2.begin(), mas2.end(), back_inserter(res));
40     printResult(res);
41 }
42
43 void sym(vector<int>& mas1, vector<int>& mas2) {
44     vector<int> res;
45     sort(mas1.begin(), mas1.end());
46     sort(mas2.begin(), mas2.end());
47
48     set_symmetric_difference(mas1.begin(), mas1.end(), mas2.begin(), mas2.end(), back_inserter(res));
49     printResult(res);
50 }
51
```

```
51
52 int main() {
53     int N, M;
54
55     cin >> N;
56     if (N < 1 || N > 100) {
57         return 1;
58     }
59
60     vector<int> mas1(N);
61     for (int i = 0; i < N; i++) {
62         cin >> mas1[i];
63         if (mas1[i] < 1 || mas1[i] > 100) {
64             return 1;
65         }
66     }
67
68     cin >> M;
69     if (M < 1 || M > 100) {
70         return 1;
71     }
72
73     vector<int> mas2(M);
74     for (int i = 0; i < M; i++) {
75         cin >> mas2[i];
76         if (mas2[i] < 1 || mas2[i] > 100) {
77             return 1;
78         }
79     }
80
81     vidn(mas1, mas2);
82     vidn(mas2, mas1);
83     per(mas1, mas2);
84     obj(mas1, mas2);
85     sym(mas1, mas2);
86
87     return 0;
88 }
89
```

```

5
1 2 3 4 5
5
4 5 6 7 8
3
1 2 3

3
6 7 8

2
4 5

8
1 2 3 4 5 6 7 8

6
1 2 3 6 7 8

```

Б) з stl

```

1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5
6  void printResult(const vector<int>& res) {
7      cout << res.size() << endl;
8      for (int num : res) {
9          cout << num << " ";
10     }
11     cout << endl << endl;
12 }
13
14 void vidn(const vector<int>& mas1, const vector<int>& mas2) {
15     vector<int> res;
16     vector<int> sm1 = mas1;
17     vector<int> sm2 = mas2;
18
19     sort(sm1.begin(), sm1.end());
20     sort(sm2.begin(), sm2.end());
21
22     int i = 0, j = 0;
23     while (i < sm1.size() && j < sm2.size()) {
24         if (sm1[i] < sm2[j]) {
25             res.push_back(sm1[i]);
26             i++;
27         } else if (sm1[i] > sm2[j]) {
28             j++;
29         } else {
30             i++;
31             j++;
32         }
33     }
34
35     while (i < sm1.size()) {
36         res.push_back(sm1[i++]);
37     }
38
39     printResult(res);
40 }
41
42 void per(const vector<int>& mas1, const vector<int>& mas2) {
43     vector<int> res;
44     vector<int> sm1 = mas1;
45     vector<int> sm2 = mas2;
46
47     sort(sm1.begin(), sm1.end());
48     sort(sm2.begin(), sm2.end());
49
50     int i = 0, j = 0;
51     while (i < sm1.size() && j < sm2.size()) {
52         if (sm1[i] < sm2[j]) {
53             i++;
54         } else if (sm1[i] > sm2[j]) {
55             j++;

```

```

56     } else {
57         res.push_back(sm1[i]);
58         i++;
59         j++;
60     }
61 }
62
63 printResult(res);
64 }
65
66 void obj(const vector<int>& mas1, const vector<int>& mas2) {
67     vector<int> res;
68     vector<int> sm1 = mas1;
69     vector<int> sm2 = mas2;
70
71     sort(sm1.begin(), sm1.end());
72     sort(sm2.begin(), sm2.end());
73
74     int i = 0, j = 0;
75     while (i < sm1.size() && j < sm2.size()) {
76         if (sm1[i] < sm2[j]) {
77             res.push_back(sm1[i++]);
78         } else if (sm1[i] > sm2[j]) {
79             res.push_back(sm2[j++]);
80         } else {
81             res.push_back(sm1[i]);
82             i++;
83             j++;
84         }
85     }
86
87     while (i < sm1.size()) {
88         res.push_back(sm1[i++]);
89     }
90
91     while (j < sm2.size()) {
92         res.push_back(sm2[j++]);
93     }
94
95     printResult(res);
96 }
97
98 void sym(const vector<int>& mas1, const vector<int>& mas2) {
99     vector<int> res;
100    vector<int> sm1 = mas1;
101    vector<int> sm2 = mas2;
102
103    sort(sm1.begin(), sm1.end());
104    sort(sm2.begin(), sm2.end());
105

```

```

105
106     int i = 0, j = 0;
107     while (i < sm1.size() && j < sm2.size()) {
108         if (sm1[i] < sm2[j]) {
109             res.push_back(sm1[i++]);
110         } else if (sm1[i] > sm2[j]) {
111             res.push_back(sm2[j++]);
112         } else {
113             i++;
114             j++;
115         }
116     }
117
118     while (i < sm1.size()) {
119         res.push_back(sm1[i++]);
120     }
121
122     while (j < sm2.size()) {
123         res.push_back(sm2[j++]);
124     }
125
126     printResult(res);
127 }
128
129 int main() {
130     int N, M;
131
132     cin >> N;
133     if (N < 1 || N > 100) {
134         return 1;
135     }
136
137     vector<int> mas1(N);
138     for (int i = 0; i < N; i++) {
139         cin >> mas1[i];
140         if (mas1[i] < 1 || mas1[i] > 100) {
141             return 1;
142         }
143     }
144
145     cin >> M;
146     if (M < 1 || M > 100) {
147         return 1;
148     }
149
150     vector<int> mas2(M);
151     for (int i = 0; i < M; i++) {
152         cin >> mas2[i];
153         if (mas2[i] < 1 || mas2[i] > 100) {
154             return 1;
155         }
156     }
157
158     vidn(mas1, mas2);
159     vidn(mas2, mas1);
160     per(mas1, mas2);
161     obj(mas1, mas2);
162     sym(mas1, mas2);
163
164     return 0;
165 }
166

```

```

5
1 2 3 4 5
5
4 5 6 7 8
3
1 2 3

3
6 7 8

2
4 5

8
1 2 3 4 5 6 7 8

6
1 2 3 6 7 8

```

Task 7 - Lab# programming: Algotester Lab 6 (1 год 30 хв)

У вас є шахова дошка розміром 8×8 та дуже багато фігур.

Кожна клітинка може мати таке значення:

- Пуста клітинка O
- Пішак P
- Тура R
- Кінь N
- Слон B
- Король K
- Королева Q

Вам дають позиції фігур на дошці (всі фігури одного кольору, кількість королів може бути > 1).

Далі йдуть Q запитів з координатами клітинки $\{x, y\}$. На кожен запит ви маєте вивести стрічку s_i - посортовані за алфавітом букви фігур, які атакують цю клітинку (пішаки атакують вниз).

У випадку, якщо на клітинці стоїть якась фігура - виведіть символ X .

У випадку, якщо клітинку не атакують - виведіть O .

Наявність фігури у певній клітинці не блокує атаку для іншої фігури. Тобто якщо між турою та клітинкою стоїть інша фігура - вважається що тура атакує цю клітинку.

16 годин тому	C++ 23	Зараховано	0.003	1.332	Перегляд
---------------	--------	------------	-------	-------	--------------------------

```

1 #include <iostream>
2 #include <vector>
3 #include <cmath>
4 #include <algorithm>
5
6 using namespace std;
7
8 struct Figure {
9     char name;
10    int x;
11    int y;
12 };
13
14 const vector<char> figuresType = {'P', 'R', 'N', 'B', 'K', 'Q', 'X', 'O'};
15
16 int cellCheck(const Figure &f, int x, int y) {
17     if (f.x == x && f.y == y) return 0; // Клітинка зайнята
18     int dx = abs(f.x - x);
19     int dy = abs(f.y - y);
20     if (f.name == 'P' && f.x == x - 1 && dy == 1) return 0;
21     if (f.name == 'R' && (f.x == x || f.y == y)) return 1;
22     if (f.name == 'N' && ((dx == 2 && dy == 1) || (dx == 1 && dy == 2))) return 2;
23     if (f.name == 'B' && dx == dy) return 3;
24     if (f.name == 'K' && dx <= 1 && dy <= 1) return 4;
25     if (f.name == 'Q' && (f.x == x || f.y == y || dx == dy)) return 5;
26     return -1;
27 }
28
29 int main() {
30     vector<Figure> figures;
31
32     for (int i = 0; i < 8; i++) {
33         string row;
34         cin >> row;
35         for (int j = 0; j < row.size(); j++) {
36             if (row[j] != 'O') {
37                 figures.push_back({row[j], i + 1, j + 1});
38             }
39         }
40     }
41
42     int Q;
43     cin >> Q;
44     vector<vector<char>> answers(Q);
45
46     for (int q = 0; q < Q; q++) {
47         int x, y;
48         cin >> x >> y;
49
50         bool empty = true;
51         bool occupied = false;
52         vector<bool> found(6, false);
53
54         for (const auto &figure : figures) {
55             int index = cellCheck(figure, x, y);

```

```

55         int index = cellCheck(figure, x, y);
56         if (index == 6) {
57             occupied = true;
58             break;
59         }
60         if (index >= 0 && index <= 5) {
61             found[index] = true;
62             empty = false;
63         }
64     }
65
66     if (occupied) {
67         answers[q].push_back('X');
68     } else if (empty) {
69         answers[q].push_back('O');
70     } else {
71         for (int i = 0; i < 6; i++) {
72             if (found[i]) {
73                 answers[q].push_back(figuresType[i]);
74             }
75         }
76         sort(answers[q].begin(), answers[q].end());
77     }
78 }
79
80
81 for (const auto &answer : answers) {
82     for (char c : answer) {
83         cout << c;
84     }
85     cout << endl;
86 }
87
88 return 0;
89 }
90

```

```

00000000
R0000000
B0000000
K0000000
O0N00000
Q0000000
O0N00000
O0NOK000
3
1 2
7 7
6 6
R
O
NQ

```

Task 8 - Practice# programming: Class Practice Task (30 хв)


```

1  #include <iostream>
2  #include <fstream>
3
4  enum FileOpResult { Success, Failure };
5
6  FileOpResult write_to_file(const char *name, const char *content) {
7      std::ofstream file(name);
8
9      if (!file.is_open()) {
10         return Failure;
11     }
12     file << content;
13
14     if (!file) {
15         return Failure;
16     }
17
18     file.close();
19
20     if (!file) {
21         return Failure;
22     }
23     return Success;
24 }
25
26 FileOpResult copy_file(const char *file_from, const char *file_to) {
27     std::ifstream source(file_from, std::ios::binary);
28     if (!source.is_open()) {
29         return Failure;
30     }
31
32     std::ofstream destination(file_to, std::ios::binary);
33     if (!destination.is_open()) {
34         return Failure;
35     }
36
37     destination << source.rdbuf();
38
39     if (!destination) {
40         return Failure;
41     }
42
43     source.close();
44     destination.close();
45
46     if (!source || !destination) {
47         return Failure;
48     }
49
50     return Success;

```

```

}

int main() {
    char *filename = "example.txt";

    char *content;
    fgets(content, 100, stdin);

    FileOpResult writeResult = write_to_file(filename, content);
    if (writeResult == Success) {
        std::cout << "файл успішно створено і записано" << std::endl;
    } else {
        std::cout << "помилка при створенні або записі файлу" << std::endl;
    }

    const char *source_file = "example.txt";
    const char *destination_file = "copied.txt";

    FileOpResult copyResult = copy_file(source_file, destination_file);
    if (copyResult == Success) {
        std::cout << "скопійовано" << std::endl;
    } else {
        std::cout << "помилка при копіюванні" << std::endl;
    }

    return 0;
}

```

```

Введіть вміст файлу: alalalal fafafaafa ratatatatta
файл успішно створено і записано
файл успішно скопійовано

```

```

epic_5 > ≡ example.txt
1  alalalal fafafaafa ratatatatta
2

```

```

epic_5 > ≡ copied.txt
1  alalalal fafafaafa ratatatatta
2

```

Task 9 - Practice# programming: Self Practice Task (30 хв)

Знайди паліндром

Обмеження: 2 сек., 256 МБ

Навіть новонародженим відомо, що паліндромом називається такий рядок з букв, який читається однаково як зліва направо, так і справа наліво. Наприклад, слова **madam**, **anna** та **racecar** є паліндромами, а слова **elephant** та **penguin** не є.

Зеніку задано рядок s , який складається з n маленьких англійських літер.

Ваша задача — допомогти йому порахувати довжину найбільшого підрядка, який є паліндромом.

Вхідні дані

У першому рядку задано одне ціле число n — розмір рядка.

У другому задано рядок s , який складається з n маленьких англійських літер.

Вихідні дані

У єдиному рядку виведіть одне ціле число — довжину найбільшого підрядка рядка s , який є паліндромом.

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5
6  int expandAroundCenter(const string &s, int left, int right) {
7      while (left >= 0 && right < s.size() && s[left] == s[right]) {
8          left--;
9          right++;
10     }
11     return right - left - 1;
12 }
13
14 int main() {
15     int n;
16     string s;
17
18     cin >> n;
19     cin >> s;
20
21     int maxLength = 0;
22
23     for (int i = 0; i < n; i++) {
24
25         int len1 = expandAroundCenter(s, i, i);
26         int len2 = expandAroundCenter(s, i, i + 1);
27         maxLength = max(maxLength, max(len1, len2));
28     }
29
30     cout << maxLength << endl;
31     return 0;
32 }
33

```

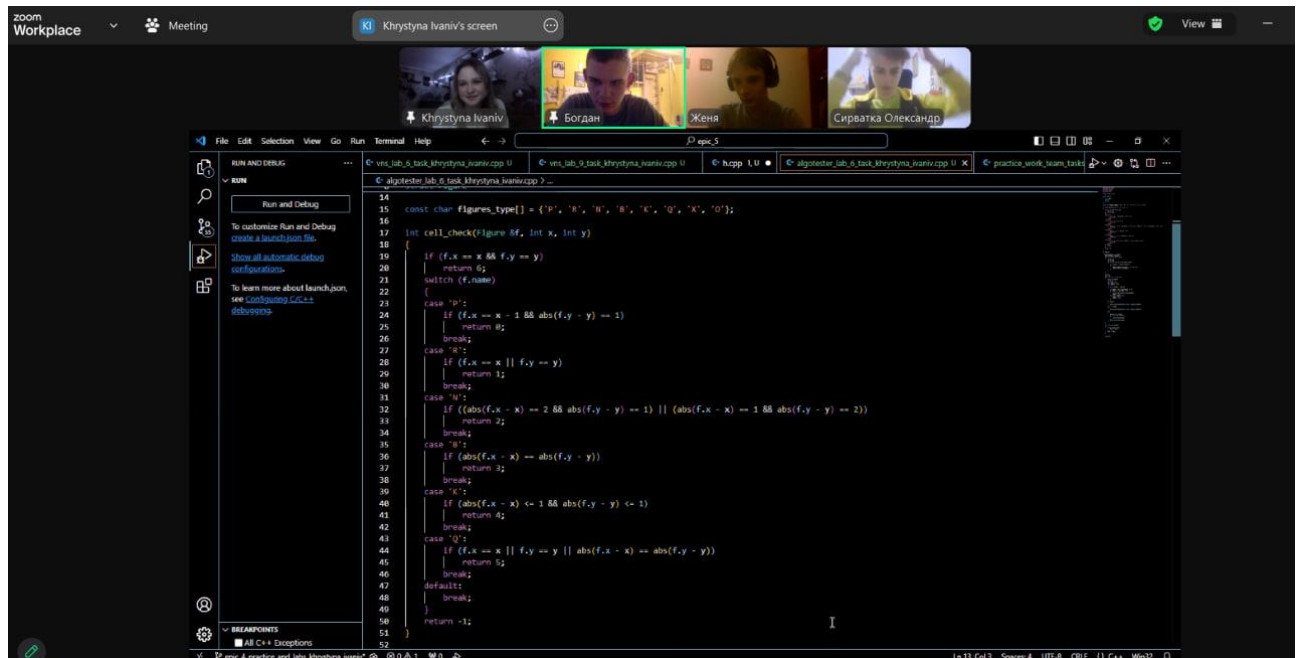
```

7
araarar
6

```

Зустрічі з командою

З однокласниками обговорювали питання по епіку онлайн, а також офлайн між парами.



Висновок:

Я навчився працювати з файлами у С++, зчитувати та записувати дані, отримав практичні навички роботи з файлами та рядковими змінними. Я ознайомився зі Стандартною бібліотекою та навчився нею користуватись. Також я навчився створювати та використовувати власні бібліотеки.