

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



Звіт

про виконання лабораторних та практичних робіт блоку № 2

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 № 6,8,9

Алготестер Лабораторної Роботи № 4,6

Практичних Робіт до блоку № 5

Виконала:

Студентка групи ШІ-12

Олійник Божена

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

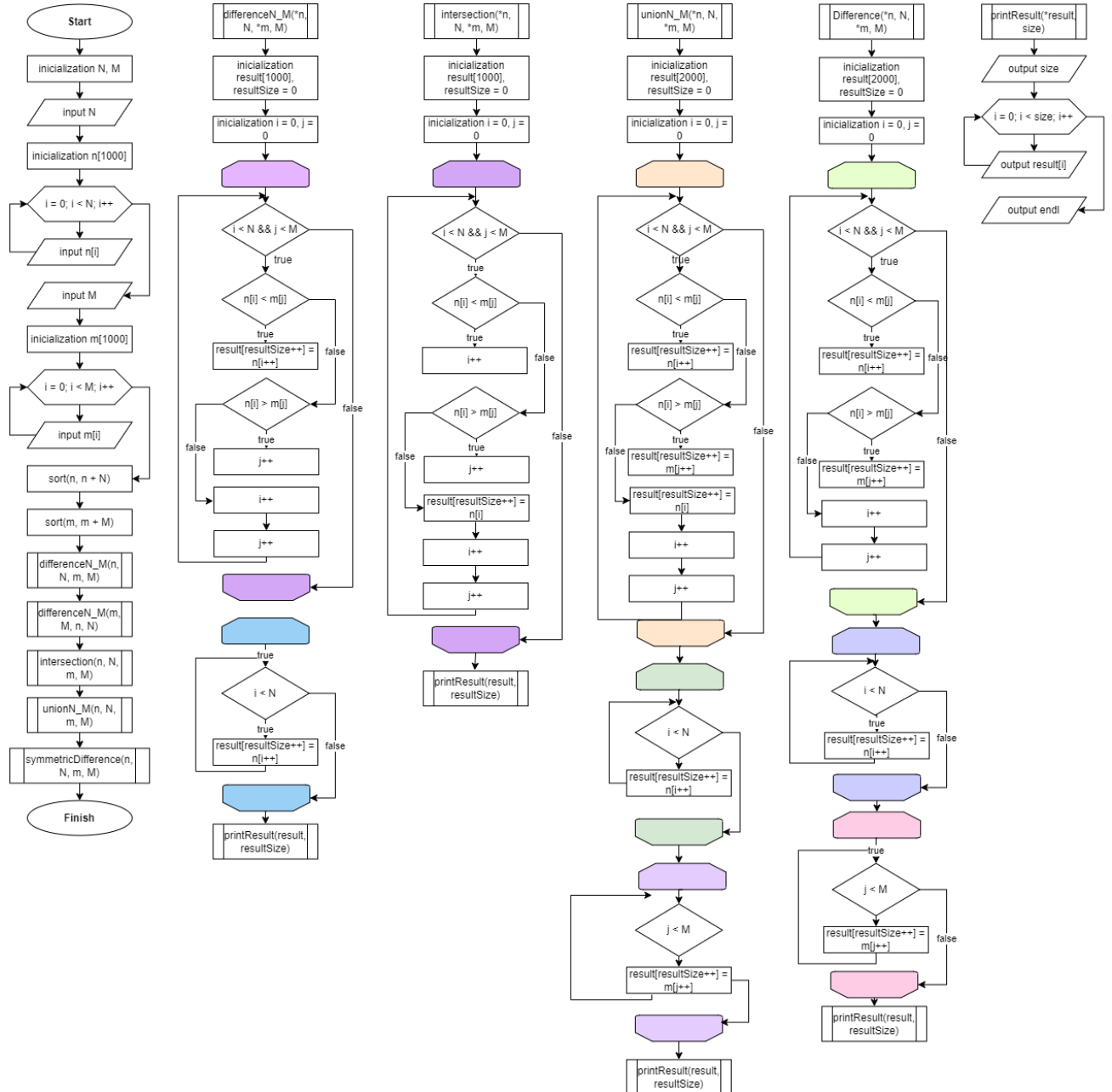
1. Навчитись працювати з файлами у C++, зокрема створювати, відкривати, читати, записувати та закривати як текстові, так і бінарні файли.
2. Ознайомитись із використанням символів і рядкових змінних при обробці текстової інформації у файлах.
3. Засвоїти основи роботи зі стандартною бібліотекою C++ для роботи з файлами, включаючи класи `fstream`, `ifstream`, `ofstream`, та їхні методи.
4. Навчитись ефективно застосовувати деталі роботи з файлами, такі як управління позиціями вказівника читання/запису, перевірка стану файлів та обробка помилок.
5. Розробити навички створення власних бібліотек для автоматизації роботи з файлами, що забезпечують зручність і повторне використання коду в інших проєктах.
6. Поглибити розуміння принципів роботи з текстовими та бінарними файлами для обробки великих обсягів даних у різних форматах.

Теоретичні відомості

- 1) Файли: <https://acode.com.ua/urok-220-bazovyj-fajlovyj-vvid-i-vyvid/>
<https://acode.com.ua/urok-221-randomnyj-fajlovyj-vvid-i-vyvid/>
- 2) Рядки: <https://acode.com.ua/urok-221-randomnyj-fajlovyj-vvid-i-vyvid/>
<https://acode.com.ua/urok-60-vvedennya-v-std-string/>
<https://acode.com.ua/urok-82-ryadky-c-style/>
- 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-iterator-stl/>
<https://acode.com.ua/urok-207-algorytmy-stl/>

Виконання роботи

Task 2 - Requirements management (understand tasks) and design activities
(draw flow diagrams and estimate tasks 3-7) (1 год 40хв)



Task 3 - Lab# programming: VNS Lab 6 (20xв)

```
1  #include <stdio.h>
2  #include <cstring>
3  #include <iostream>
4
5  using namespace std;
6
7  void convertToWords(char *sent, char words[][255], int &wordCount)
8  {
9      char *w = strtok(sent, " ");
10     wordCount = 0;
11
12     while (w != nullptr)
13     {
14         strcpy(words[wordCount++], w);
15         w = strtok(nullptr, " ");
16     }
17 }
18 int main()
19 {
20     char sentence[256];
21
22     cout << "Введіть рядок: ";
23     fgets(sentence, sizeof(sentence), stdin);
24
25     int length = sizeof(sentence);
26     if (length > 0 && sentence[length - 1] == '.')
27     {
28         sentence[length - 1] = '\0';
29     }
30
31     char words[50][255];
32     int wordCount = 0;
33
34     convertToWords(sentence, words, wordCount);
35     cout << "Слова, що співпадають з першим словом (" << words[0] << "):" << endl;
36     for (int i = 1; i < wordCount; i++)
37     {
38         if (strcmp(words[0], words[i]) == 0)
39         {
40             cout << words[i] << endl;
41         }
42     }
43
44     return 0;
45 }
46
```

```
Введіть рядок: день сьогодні був дуже гарний день
Слова, що співпадають з першим словом (день):
день
```

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

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  using namespace std;
6
7  struct Human
8  {
9      string initials;
10     string address;
11     long long phone;
12     int age;
13 };
14
15 void formFile(const char *filename)
16 {
17     ofstream file(filename, ios::binary);
18     if (file.is_open())
19     {
20         Human person[] = {
21             {"Serhiy S. O.", "Shevchenka", 380954862571, 50},
22             {"Maria O. R.", "Bandery", 380501486571, 20},
23             {"Arsen B. B.", "Gorbachevskogo", 380965874855, 14},
24             {"Matviy T. A.", "Naukova", 380954887215, 42},
25             {"Juliett N. K.", "Struyska", 380501452369, 33},
26         };
27
28         for (const auto &human : person)
29         {
30             file << "Initials: " << human.initials << "\n";
31             file << "Address: " << human.address << "\n";
32             file << "Phone number: " << human.phone << "\n";
33             file << "Age: " << human.age << "\n";
34             file << "-----\n";
35         }
36
37         file.close();
38     }
39     else
40     {
41         cout << "Помилка в відкритті файлу!" << endl;
42     }
43 }
44
45 void printFile(const char *filename)
46 {
47     ifstream file(filename, ios::binary);
48     if (file.is_open())
49     {
50         string line;
51         while (getline(file, line))
52         {
53             cout << line << endl;
54         }
55         file.close();
56     }
57     else
58     {
59         cout << "Помилка в відкритті файлу!" << endl;
60     }
61 }
62
63 void deleteAge(const char *filename, int targetAge)
64 {
65     ifstream file(filename);
66     ofstream tempFile("temp.txt");
67
68     if (file.is_open() && tempFile.is_open())
69     {
70         string line;
71         bool isDeleting = false;
72
73         while (getline(file, line))
74         {
75             if (line.find("Age: ") != string::npos)
76             {
77                 int age = stoi(line.substr(line.find(": ") + 2));
78                 if (age == targetAge)
79                 {
80                     tempFile << "-----\n";
81                     while (getline(file, line) && line != "-----")
82                     {
83                         ;
84                         continue;
85                     }
86                 }
87                 tempFile << line << "\n";
88             }
89         }
90
91         file.close();
92         tempFile.close();
93         remove(filename);
94         rename("temp.txt", filename);
95     }
96     else
97     {
98         cout << "Помилка в відкритті файлу!" << endl;
99     }
100 }
101
102 void addStatus(const char *filename, const string &targetPhone, const string &status)
103 {
104     ifstream file(filename);
105     ofstream tempFile("temp.txt");
106
107     if (file.is_open() && tempFile.is_open())
108     {
109

```

```

108     if (file.is_open() && tempFile.is_open())
109     {
110         while (getline(file, line))
111         {
112             tempFile << line << "\n";
113
114             if (line.find("Phone number: ") != string::npos)
115             {
116                 string phone = line.substr(line.find(":") + 2);
117                 if (phone == targetPhone)
118                 {
119                     addStatusAfterPhone = true;
120                 }
121             }
122
123             if (addStatusAfterPhone && line == "-----")
124             {
125                 tempFile << "Note: " << status << endl;
126                 tempFile << "-----\n";
127                 addStatusAfterPhone = false;
128             }
129         }
130
131         file.close();
132         tempFile.close();
133         remove(filename);
134         rename("temp.txt", filename);
135     }
136     else
137     {
138         cout << "Помилка в відкритті файлу!" << endl;
139     }
140 }
141
142 int main()
143 {
144     const char *filename = "humans.txt";
145     formFile(filename);
146     cout << "Initial file content:" << endl << endl;
147     printFile(filename);
148
149     int targetAge = 20;
150     cout << "\nDeleting person with age " << targetAge << endl << endl;
151     deleteAge(filename, targetAge);
152     printFile(filename);
153
154     string phone = "380501452369";
155     cout << "\nAdding status after person with that phone number " << phone << endl << endl;
156     addStatus(filename, phone, "мама");
157     printFile(filename);
158
159     return 0;
160 }

```

Initial file content:	Deleting person with age 20	Adding status after person with that phone number 380501452369
Initials: Serhiy S. O. Address: Shevchenka Phone number: 380954862571 Age: 50 -----	Initials: Serhiy S. O. Address: Shevchenka Phone number: 380954862571 Age: 50 -----	Initials: Serhiy S. O. Address: Shevchenka Phone number: 380954862571 Age: 50 -----
Initials: Maria O. R. Address: Bandery Phone number: 380501486571 Age: 20 -----	Initials: Maria O. R. Address: Bandery Phone number: 380501486571 Age: 20 -----	Initials: Maria O. R. Address: Bandery Phone number: 380501486571 Age: 20 -----
Initials: Arsen B. B. Address: Gorbachevskogo Phone number: 380965874855 Age: 14 -----	Initials: Arsen B. B. Address: Gorbachevskogo Phone number: 380965874855 Age: 14 -----	Initials: Arsen B. B. Address: Gorbachevskogo Phone number: 380965874855 Age: 14 -----
Initials: Matviy T. A. Address: Naukova Phone number: 380954887215 Age: 42 -----	Initials: Matviy T. A. Address: Naukova Phone number: 380954887215 Age: 42 -----	Initials: Matviy T. A. Address: Naukova Phone number: 380954887215 Age: 42 -----
Initials: Juliett N. K. Address: Struyska Phone number: 380501452369 Age: 33 -----	Initials: Juliett N. K. Address: Struyska Phone number: 380501452369 Age: 33 -----	Initials: Juliett N. K. Address: Struyska Phone number: 380501452369 Age: 33 -----
		Note: МАМА

Task 5 - Lab# programming: VNS Lab 9 (40хв)

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4
5  using namespace std;
6
7  int lastWordLength(const string &line)
8  {
9      int length = 0;
10     int i = line.size() - 1;
11
12     while (i >= 0 && isspace(line[i]))
13     {
14         i--;
15     }
16
17     while (i >= 0 && !isspace(line[i]))
18     {
19         length++;
20         i--;
21     }
22
23     return length;
24 }
25
26 int main()
27 {
28     ofstream f1("F1.txt");
29     if (f1.is_open())
30     {
31         f1 << "Перший рядок.\n";
32         f1 << "Другий рядок.\n";
33         f1 << "Третій рядок.\n";
34         f1 << "Четвертий рядок.\n";
35         f1 << "П'ятий рядок.\n";
36         f1 << "Шостий рядок.\n";
37         f1 << "Сьомий рядок.\n";
38         f1 << "Восьмий рядок.\n";
39         f1 << "Дев'ятий рядок.\n";
40         f1 << "Десятий рядок.\n";
41         f1.close();
42     }
43
44     ifstream inputFile("F1.txt");
45     ofstream outputFile("F2.txt");
46     string line;
47     if (inputFile.is_open() && outputFile.is_open())
48     {
49         int lineCount = 0;
50         while (getline(inputFile, line))
51         {
52             lineCount++;
53             if (lineCount >= 4)
54             {
55                 outputFile << line << endl;
56             }
57         }
58
59         inputFile.close();
60         outputFile.close();
61     }
62
63     ifstream f2("F2.txt");
64     string lastline;
65     if (f2.is_open())
66     {
67         while (getline(f2, line))
68         {
69             lastline = line;
70         }
71         f2.close();
72     }
73
74     int lastWord = lastWordLength(lastline);
75     cout << "Кількість символів в останньому слові F2: " << lastWord << endl;
76
77     return 0;
78 }

```

```

Input line № 1 : today
Input line № 2 : is
Input line № 3 : november
Input line № 4 : 21
Input line № 5 : of 2024
Input line № 6 : that day
Input line № 7 : is also known
Input line № 8 : as Day of digniry and freedom
Input line № 9 : but
Input line № 10 : it is also my birthday
Copied successfully!
Number of symbols in the last word of F2: 8
Files "F1.txt" and "F2.txt" are removed successfully.

```

Task 6.1 - Lab# programming: Algotester Lab 4 Var 3 (first task) (20 хв)

```

6  int main()
7
8      int N;
9      cin >> N;
10
11     int n[N];
12     for (int i = 0; i < N; i++)
13     {
14         cin >> n[i];
15     }
16
17     int zero[N], one[N], two[N];
18     int z = 0, o = 0, t = 0;
19
20     for (int i = 0; i < N; i++)
21     {
22         if (n[i] % 3 == 0)
23         {
24             zero[z++] = n[i];
25         }
26         else if (n[i] % 3 == 1)
27         {
28             one[o++] = n[i];
29         }
30         else
31         {
32             two[t++] = n[i];
33         }
34     }
35
36     sort(zero, zero + z);
37     sort(one, one + o, greater<int>());
38     sort(two, two + t);
39
40     int result[N], count = 0;
41
42     for (int i = 0; i < z; i++)
43     {
44         result[count++] = zero[i];
45     }
46     for (int i = 0; i < o; i++)
47     {
48         result[count++] = one[i];
49     }
50     for (int i = 0; i < t; i++)
51     {
52         result[count++] = two[i];
53     }
54
55     count = unique(result, result + count) - result;
56
57     cout << count << endl;
58     for (int i = 0; i < count; i++)
59     {
60         cout << result[i] << " ";
61     }

```

```

10
1 33 4 8 6 5 2 7 5 0
9
0 6 33 7 4 1 2 5 8

```

Task 6.2 - Lab# programming: Algotester Lab 4 Var 1 (second task) (2 год)


```

1 #include <iostream>
2 #include <algorithm>
3 using namespace std;
4
5 void printResult(const int *result, int size) {
6     cout << size << endl;
7     for (int i = 0; i < size; i++) {
8         cout << result[i] << " ";
9     }
10    cout << endl;
11 }
12
13 void differenceN_M(const int *n, int N, const int *m, int M) {
14     int result[1000], resultSize = 0;
15
16     int i = 0, j = 0;
17     while (i < N && j < M) {
18         if (n[i] < m[j]) {
19             result[resultSize++] = n[i++];
20         } else if (n[i] > m[j]) {
21             j++;
22         } else {
23             i++;
24             j++;
25         }
26     }
27
28     while (i < N) {
29         result[resultSize++] = n[i++];
30     }
31
32     printResult(result, resultSize);
33 }
34
35 void intersection(const int *n, int N, const int *m, int M) {
36     int result[1000], resultSize = 0;
37
38     int i = 0, j = 0;
39     while (i < N && j < M) {
40         if (n[i] < m[j]) {
41             i++;
42         } else if (n[i] > m[j]) {
43             j++;
44         } else {
45             result[resultSize++] = n[i];

```

```

81 void symmetricDifference(const int *n, int N, const int *m, int M) {
82     int result[2000], resultSize = 0;
83
84     int i = 0, j = 0;
85     while (i < N && j < M) {
86         if (n[i] < m[j]) {
87             result[resultSize++] = n[i++];
88         } else if (n[i] > m[j]) {
89             result[resultSize++] = m[j++];
90         } else {
91             i++;
92             j++;
93         }
94     }
95
96     while (i < N) {
97         result[resultSize++] = n[i++];
98     }
99
100    while (j < M) {
101        result[resultSize++] = m[j++];
102    }
103
104    printResult(result, resultSize);
105 }
106
107 int main() {
108     int N, M;
109
110     cin >> N;
111     int n[1000];
112     for (int i = 0; i < N; ++i) {
113         cin >> n[i];
114     }
115
116     cin >> M;
117     int m[1000];
118     for (int i = 0; i < M; ++i) {
119         cin >> m[i];
120     }
121
122     sort(n, n + N);
123     sort(m, m + M);
124
125     differenceN_M(n, N, m, M);

```

```

int main() {
    int N, M;

    cin >> N;
    int n[1000];
    for (int i = 0; i < N; ++i) {
        cin >> n[i];
    }

    cin >> M;
    int m[1000];
    for (int i = 0; i < M; ++i) {
        cin >> m[i];
    }

    sort(n, n + N);
    sort(m, m + M);

    differenceN_M(n, N, m, M);
    cout << endl;

    differenceN_M(m, M, n, N);
    cout << endl;

    intersection(n, N, m, M);
    cout << endl;

    unionN_M(n, N, m, M);
    cout << endl;

    symmetricDifference(n, N, m, M);
    cout << endl;

    return 0;
}

```

```

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 20 хв)

```

1  #include <iostream>
2  #include <cmath>
3  #include <cstring>
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 char figures_type[] = {'P', 'R', 'N', 'B', 'K', 'Q', 'X', 'O'};
15
16 int cell_check(Figure &f, int x, int y) {
17     if (f.x == x && f.y == y)
18         return 6;
19     switch (f.name) {
20     case 'P':
21         if (f.x == x - 1 && abs(f.y - y) == 1)
22             return 0;
23         break;
24     case 'R':
25         if (f.x == x || f.y == y)
26             return 1;
27         break;
28     case 'N':
29         if ((abs(f.x - x) == 2 && abs(f.y - y) == 1) || (abs(f.x - x) == 1 && abs(f.y - y) == 2))
30             return 2;
31         break;
32     case 'B':
33         if (abs(f.x - x) == abs(f.y - y))
34             return 3;
35         break;
36     case 'K':
37         if (abs(f.x - x) <= 1 && abs(f.y - y) <= 1)
38             return 4;
39         break;
40     case 'Q':
41         if (f.x == x || f.y == y || abs(f.x - x) == abs(f.y - y))
42             return 5;
43         break;
44     default:
45         break;
46     }
47     return -1;
48 }
49
50 int main() {
51     Figure figures[64];
52     int figures_count = 0;
53
54     for (int i = 0; i < 8; i++) {
55         string row;

```

```

50 int main() {
51     for (int i = 0; i < 8; i++) {
52         cin >> row;
53         for (int j = 0; j < row.size(); j++) {
54             if (row[j] != figures_type[7]) {
55                 figures[figures_count++] = {row[j], i + 1, j + 1};
56             }
57         }
58     }
59
60     int Q;
61     cin >> Q;
62     char answer[100][7];
63     int answer_sizes[100] = {0};
64
65     for (int q = 0; q < Q; q++) {
66         int x, y;
67         cin >> x >> y;
68
69         bool case1 = true;
70         bool case2 = false;
71         bool found[6] = {false};
72
73         for (int i = 0; i < figures_count; i++) {
74             int index = cell_check(figures[i], x, y);
75             if (index >= 0 && index <= 5) {
76                 found[index] = true;
77                 case1 = false;
78             } else if (index == 6) {
79                 case1 = false;
80                 case2 = true;
81             }
82         }
83
84         if (case2) {
85             answer[q][0] = figures_type[6];
86             answer_sizes[q] = 1;
87         } else if (case1) {
88             answer[q][0] = figures_type[7];
89             answer_sizes[q] = 1;
90         } else {
91             int size = 0;
92             for (int i = 0; i < 6; i++) {
93                 if (found[i]) {
94                     answer[q][size++] = figures_type[i];
95                 }
96             }
97             answer_sizes[q] = size;
98             sort(answer[q], answer[q] + size);
99         }
100     }
101
102     for (int q = 0; q < Q; q++) {
103         for (int i = 0; i < answer_sizes[q]; i++) {
104             cout << answer[q][i];
105         }
106     }
107 }

```

```

00000000
0R000000
00N00000
0000P000
00000000
00000000
K0Q00000
0000000R
7
8 1
1 2
5 4
5 1
6 2
8 4
6 7
KR
NR
NP
Q
KQR
QR
0

```

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

```

1  #include <iostream>
2  #include <algorithm>
3  #include <fstream>
4  #include <string>
5
6  using namespace std;
7
8  void writeToFile(const string &name, const string &content)
9  {
10     ofstream file(name, ios::out);
11     if (file.is_open())
12     {
13         file << content;
14         file.close();
15         cout << "Success in creating and opening file \"" << name << "\" << endl;
16     }
17     else
18     {
19         cout << "Failure in creating or opening file \"" << name << "\" << endl;
20     }
21
22     ifstream infile(name);
23     if (infile.is_open())
24     {
25         string line;
26         getline(infile, line);
27
28         cout << line << endl;
29
30         infile.close();
31         cout << "Success in reading from file \"" << name << "\" << endl;
32     }
33     else
34     {
35         cout << "Failure in reading from file \"" << name << "\" << endl;
36     }
37 }
38
39 void copyFile (const string &file_from, const string &file_to){
40     ifstream infile(file_from);
41     ofstream outfile(file_to, ios::out);
42
43     if (infile.is_open() && outfile.is_open())
44     {
45         string line;
46         getline(infile, line);
47
48         outfile << line;
49
50         infile.close();
51         cout << "Success in copying from \"" << file_from << "\" to \"" << file_to << "\" << endl;
52     }
53     else
54     {

```

```

55         cout << "Failure in copying from \"" << file_from << "\" to \"" << file_to << "\" << endl;
56     }
57 }
58 }
59
60 int main()
61 {
62     string filename;
63     cout << "Input the name of file: ";
64     cin >> filename;
65
66     cin.ignore();
67
68     string content;
69     cout << "Input content of file: ";
70     getline(cin, content);
71
72     string copyfilename;
73     cout << "Input the name of file in which we'll copy information: ";
74     cin >> copyfilename;
75
76     writeToFile(filename, content);
77     copyFile(filename, copyfilename);
78
79     if (remove(filename.c_str()) == 0 && remove(copyfilename.c_str()) == 0)
80     {
81         cout << "Файл \"" << filename << "\", \"" << copyfilename << "\" удалено << endl;
82     }
83     else
84     {
85         cout << "Не удалось удалить файл \"" << filename << "\", \"" << copyfilename << "\" << endl;
86     }
87
88     return 0;
89 }

```

```

Input the name of file: original
Input content of file: this text will be copied
Input the name of file in which we'll copy information: copy
Success in creating and opening file "original"
this text will be copied
Success in reading from file "original"
Success in copying from "original" to "copy"
Files "original" , "copy" are removed

```

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

```

1  #include <iostream>
2  #include <algorithm>
3  #include <unordered_set>
4  using namespace std;
5
6  int main()
7  {
8      int n, k;
9      cin >> n >> k;
10
11     int* numbers = new int[n];
12
13     for (int i = 0; i < n; i++)
14     {
15         cin >> numbers[i];
16     }
17
18     sort(numbers, numbers + n);
19
20     int* uniqueNumbers = new int[n];
21     int newSize = 0;
22     unordered_set<int> seen;
23
24     for (int i = 0; i < n; i++)
25     {
26         if (seen.find(numbers[i]) == seen.end())
27         {
28             seen.insert(numbers[i]);
29             uniqueNumbers[newSize++] = numbers[i];
30         }
31     }
32
33     k %= newSize;
34     int* rotatedNumbers = new int[newSize];
35     for (int i = 0; i < newSize; i++)
36     {
37         rotatedNumbers[i] = uniqueNumbers[(i + k) % newSize];
38     }
39
40     cout << newSize << endl;
41     for (int i = 0; i < newSize; i++)
42     {
43         cout << rotatedNumbers[i] << " ";
44     }
45     cout << endl;
46
47     delete[] numbers;
48     delete[] uniqueNumbers;
49     delete[] rotatedNumbers;
50
51     return 0;
52 }

```

```

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

```

Зустрічі з командою

З командою зустрічалися двічі, на зустрічах обговорювали питання та прогрес по епіку.



Висновок

В ході даного етапу я навчилася працювати з файлами у C++, включаючи створення, відкриття, читання, запис та закриття текстових і бінарних файлів. Також опанувала роботу з рядковими змінними для обробки текстової інформації, класи стандартної бібліотеки `fstream`, `ifstream`,

ofstream, а також засвоїла управління позиціями вказівника читання/запису, перевірку стану файлів і обробку помилок.