

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-12

Тимчук Дмитро

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, виконувати різні операції з ними (в мові C/C++)
- 2) Використання бібліотечних функцій

Теоретичні відомості:

1. Робота з файлами C
2. Робота з файлами C++

Індивідуальний план опрацювання теорії:

- Тема №1 Робота з файлами C (50 хв)
(<https://www.youtube.com/watch?v=FeNqHytI0fA>)
- Тема №2 Робота з файлами C++ (40 хв)
(<https://www.youtube.com/watch?v=SSNJ7alki-E>)

Виконання роботи

Завдання №1 Epic 5 Task 3 - Lab# programming: VNS Lab 6

```
1  #include <bits/stdc++.h>
2  using namespace std;
3  void sortdigits(vector <string> &words){
4      vector <int> digits;
5      for(int i = 0; i < words.size(); i++){
6          string word = words[i];
7          for(char c : word){
8              if(isdigit(c)){
9                  digits.push_back(c);
10             }
11         }
12         sort(digits.rbegin(), digits.rend());
13         int index = 0;
14         for (char &c : word){
15             if(isdigit(c)){
16                 c = digits[index];
17                 index++;
18             }
19         }
20         words[i] = word;
21         digits.clear();
22     }
23 }
```

```
24 int main(){
25     char s[256];
26     gets(s);
27     string input (s);
28     vector <char> temp;
29     int j = 0;
30     vector <string> words;
31     bool ifSpace = false;
32     for(int i = 0; i < input.length(); i++){
33         if (input[i] != ' '){
34             temp.push_back(input[i]);
35             ifSpace = true;
36             continue;
37         }
38         if(ifSpace){
39             string tempstr(temp.begin(), temp.end());
40             words.push_back(tempstr);
41             j++;
42             temp.clear();
43             ifSpace = false;
44         }
45     }
46     if(temp.size() != 0){
47         string tempstr(temp.begin(), temp.end());
48         words.push_back(tempstr);
49         temp.clear();
50     }
51
52     sortdigits(words);
53     for(string d : words){
54         cout << d << " ";
55     }
56     return 0;
57 }
58 }
```

Завдання №2 Epic 5 Task 4 - Lab# programming: VNS Lab 8

```
1  #include <iostream>
2  #include<stdio.h>
3  #include<vector>
4  using namespace std;
5  struct sportTeam{
6      char teamName[20];
7      char city[20];
8      int playerQ;
9      int pointQ;
10 };
11
12 void addDataInFile(const char* fileName){
13     sportTeam arr[]{
14         {"RealMadrid", "Madrid", 15, 32},
15         {"Barcelona", "Barcelona", 18, 19},
16         {"Dynamo Kiev", "Kiev", 11, 33},
17         {"Karpaty", "Lviv", 16, 39},
18         {"Vorskla", "Poltava", 15, 14}
19     };
20     FILE* fileStream;
21     fileStream = fopen(fileName, "wb");
22     if (fileStream == nullptr){
23         cerr << "Не вдалося відкрити файл";
24         exit(0);
25     }
26     for (size_t i = 0; i < 5; i++){
27         {
28             int written = fwrite(&arr[i], sizeof(sportTeam), 1, fileStream);
29             if (written != 1){
30                 cerr << "Не вдалось записати дані";
31                 exit(0); ///?
32             }
33         }
34     }
35     fclose(fileStream);
36 }
37
38 void pirntDataFromFile(const char* fileName){
39     FILE* fileStream;
40     fileStream = fopen(fileName, "rb");
41     if (fileStream == nullptr){
42         cerr << "Не вдалося відкрити файл";
43         exit(0);
44     }
45     sportTeam temp;
46     while(fread(&temp, sizeof(sportTeam), 1, fileStream) == 1){
47         cout << "Назва команди: " << temp.teamName << endl;
48         cout << "Місто: " << temp.city << endl;
49         cout << "Кількість гравців: " << temp.playerQ << endl;
50         cout << "Кількість очок: " << temp.pointQ << endl << endl;
51     }
52     fclose(fileStream);
53 }
```

```

54 void deleteElemFromFile(const char* fileName){
55     FILE* fileStream;
56     fileStream = fopen(fileName, "rb");
57     if (fileStream == nullptr){
58         cerr << "Не вдалося відкрити файл";
59         exit(0);
60     }
61     vector<sportTeam> teams;
62     sportTeam temp;
63     int minPoint;
64     cout << "Введіть мінімальну кількість очок: ";
65     cin >> minPoint;
66     cout << endl;
67     while(fread(&temp, sizeof(sportTeam), 1, fileStream) == 1){
68         if(temp.pointQ >= minPoint){
69             teams.push_back(temp);
70         }
71     }
72     fclose(fileStream);
73     fileStream = fopen(fileName, "wb");
74     if (fileStream == nullptr){
75         cerr << "Не вдалося відкрити файл для запису";
76         exit(0);
77     }
78     for(int i = 0; i < teams.size(); i++){
79         fwrite(&teams[i], sizeof(sportTeam), 1, fileStream);
80     }
81     fclose(fileStream);
82 }

```

```

83
84 void addElemToFile(const char* fileName){
85     FILE* fileStream;
86     fileStream = fopen(fileName, "rb");
87     if(fileStream == nullptr){
88         cerr << "Не вдається відкрити файл для зчитування";
89         exit(0);
90     }
91     vector<sportTeam> newTeams{
92         {"Shakhtar", "Donetsk", 11, 35},
93         {"Chornomorets", "Odesa", 20, 9}
94     };
95     vector<sportTeam> teams = newTeams;
96     sportTeam temp;
97     while(fread(&temp, sizeof(sportTeam), 1, fileStream) == 1){
98         teams.push_back(temp);
99     }
100     fclose(fileStream);
101     fileStream = fopen(fileName, "wb");
102     if (fileStream == nullptr){
103         cerr << "Не вдалося відкрити файл";
104         exit(0);
105     }
106     for (size_t i = 0; i < teams.size(); i++)
107     {
108         fwrite(&teams[i], sizeof(sportTeam), 1, fileStream);
109     }
110     fclose(fileStream);
111 }

```

```

112
113 int main(){
114     const char* fileName = "lab8.bin";
115     addDataInFile(fileName);
116     pirntDataFromFile(fileName);
117     deleteElemFromFile(fileName);
118     addElemToFile(fileName);
119     cout << "Елементи після операцій: " << endl << endl;
120     pirntDataFromFile(fileName);
121     return 0;
122 }

```

Завдання №3 Task 5 - Lab# programming: VNS Lab 9

```
1  #include<iostream>
2  #include<stdio.h>
3  #include<vector>
4  #include<string>
5  using namespace std;
6
7
8  void inputInFile(const char* fileName1){
9      FILE* fileStream;
10     fileStream = fopen(fileName1, "w");
11     if(fileStream == nullptr){
12         cerr << "Не вдається відкрити файл";
13         exit(0);
14     }
15     string temp;
16     for(int i = 0; i < 5; i++){
17         getline(cin, temp);
18         fputs(temp.c_str(), fileStream);
19         fputs("\n", fileStream);
20     }
21     fclose(fileStream);
22 }
23
24 void copy(const char* fileName1, const char* fileName2){
25     FILE* fileStream1;
26     FILE* fileStream2;
27     fileStream1 = fopen(fileName1, "r");
28     fileStream2 = fopen(fileName2, "w");
29     if(fileStream1 == nullptr){
30         cerr << "Не вдається відкрити файл1";
31         exit(0);
32     }
33     if(fileStream2 == nullptr){
34         cerr << "Не вдається відкрити файл2";
35         fclose(fileStream1);
36         exit(0);
37     }
38     cout << "Введіть N1 та N2: ";
39     int N1, N2;
40     cin >> N1 >> N2;
41     if (N1 > N2) {
42         cout << "Невірно задані номери рядків. N1 має бути меншим за N2" << endl;
43         fclose(fileStream1);
44         fclose(fileStream2);
45         exit(0);
46     }
47     const int size = 50;
48     char str[size];
49     int current_line = 1;
50     while(fgets(str, size, fileStream1)){
51         if((str[0] == 'A' || str[0] == 'a') && current_line >= N1 && current_line <= N2){
52             fputs(str, fileStream2);
53         }
54     }
55     fclose(fileStream1);
56     fclose(fileStream2);
57 }
```

```

59 bool isConsonant (char c){
60     c = tolower(c);
61     return (c >= 'a' && c <= 'z') && !(c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' || c == 'y');
62 }
63
64 int maxConsonant(const char* fileName){
65     FILE* fileStream;
66     fileStream = fopen(fileName, "r");
67     if(fileStream == nullptr){
68         cerr << "Не вдається відкрити файл";
69         exit(0);
70     }
71     int maxConsonants = 0;
72     int lineNumber = 0;
73     int currentLine = 1;
74     int consonantCount = 0;
75     char str[50];
76     while(fgets(str, 50, fileStream)){
77         consonantCount = 0;
78         currentLine++;
79         for (size_t i = 0; str[i] != '\0'; i++)
80         {
81             if(isConsonant(str[i])){
82                 consonantCount++;
83             }
84         }
85         if (consonantCount > maxConsonants){
86             maxConsonants = consonantCount;
87             lineNumber = currentLine;
88         }
89     }
90     return lineNumber;
91 }
92
93
94 int main() {
95     const char* fileName1 = "F1.txt";
96     const char* fileName2 = "F2.txt";
97     inputInFile(fileName1);
98     copy(fileName1, fileName2);
99     cout << "Номер рядка з найбільшою кількістю приголосних - " << maxConsonant(fileName2);
100     return 0;
101 }

```

Завдання №4 Epic 5 Task 6 - Lab# programming: Algotester Lab 4

Lab4v2 (version 1)

```
1  #include<iostream>
2  #include<vector>
3  #include <algorithm>
4  using namespace std;
5
6  int main() {
7      int n, t, ost0 = 0, ost1 = 0, ost2 = 0;
8      cin >> n;
9      vector<int> vec, ostVec(n);
10     for (size_t i = 0; i < n; i++)
11     {
12         cin >> t;
13         vec.push_back(t);
14         if (t % 3 == 0){
15             ostVec.insert(ostVec.begin() + ost0, t);
16             ost0++;
17             ost1++;
18             ost2++;
19         }
20         else if (t % 3 == 1){
21             ostVec.insert(ostVec.begin() + ost1, t);
22             ost1++;
23             ost2++;
24         }
25         else{
26             ostVec.insert(ostVec.begin() + ost2, t);
27             ost2++;
28         }
29     }
30     sort(ostVec.begin(), ostVec.begin() + ost0);
31     sort(ostVec.begin() + ost0, ostVec.begin() + ost1, greater<int>());
32     sort(ostVec.begin() + ost1, ostVec.begin() + ost2);
33     auto new_end = unique(ostVec.begin(), ostVec.end());
34     ostVec.erase(new_end, ostVec.end());
35     int last = distance(ostVec.begin(), new_end) - 1;
36     cout << last << endl;
37     for (int i = 0; i < last; i++)
38     {
39         cout << ostVec[i] << " ";
40     }
41     return 0;
42 }
```


Lab4v2 (version 2)

```
1  #include<iostream>
2  #include<vector>
3  #include <algorithm>
4  using namespace std;
5
6  void quickSort(vector<int> &vec, int first, int last){
7      int middle = vec[(first + last) / 2];
8      int i = first;
9      int j = last;
10     do{
11         while(vec[i] < middle){
12             i++;
13         }
14         while(vec[j] > middle){
15             j--;
16         }
17         if (i <= j){
18             swap(vec[i], vec[j]);
19             i++;
20             j--;
21         }
22     }while(i <= j);
23     if (j > first){
24         quickSort(vec, first, j);
25     }
26     if (i < last){
27         quickSort(vec, i, last);
28     }
29 }
30
31 void quickSortDescending(vector<int> &vec, int first, int last) {
32     int middle = vec[(first + last) / 2];
33     int i = first;
34     int j = last;
35     do {
36         while (vec[i] > middle) {
37             i++;
38         }
39         while (vec[j] < middle) {
40             j--;
41         }
42         if (i <= j) {
43             swap(vec[i], vec[j]);
44             i++;
45             j--;
46         }
47     } while (i <= j);
48     if (j > first) {
49         quickSortDescending(vec, first, j);
50     }
51     if (i < last) {
52         quickSortDescending(vec, i, last);
53     }
54 }
```

```

56 void myUnique(vector<int> &vec, int &size){
57     vector<int> uniqueVec;
58     for (int num : vec){
59         bool found = false;
60         for (int unqnum : uniqueVec){
61             if (num == unqnum){
62                 found = true;
63                 size--;
64             }
65         }
66         if (!found){
67             uniqueVec.push_back(num);
68         }
69     }
70     vec = uniqueVec;
71 }
72
73 int main() {
74     int n, t;
75     cin >> n;
76     vector<int> vec, ost0Vec, ost1Vec, ost2Vec;
77     for (size_t i = 0; i < n; i++)
78     {
79         cin >> t;
80         vec.push_back(t);
81         if (t % 3 == 0){
82             ost0Vec.push_back(t);
83         }
84         else if (t % 3 == 1){
85             ost1Vec.push_back(t);
86         }
87         else{
88             ost2Vec.push_back(t);
89         }
90     }
91
92     quickSort(ost0Vec, 0, ost0Vec.size() - 1);
93     quickSortDescending(ost1Vec, 0, ost1Vec.size() - 1);
94     quickSort(ost2Vec, 0, ost2Vec.size() - 1);
95     vector<int>res;
96     res.insert(res.end(), ost0Vec.begin(), ost0Vec.end());
97     res.insert(res.end(), ost1Vec.begin(), ost1Vec.end());
98     res.insert(res.end(), ost2Vec.begin(), ost2Vec.end());
99     int sizeOst = res.size();
100     myUnique(res, sizeOst);
101     cout << sizeOst << endl;
102     for (int i = 0; i < sizeOst; i++)
103     {
104         cout << res[i] << " ";
105     }
106     return 0;
107 }

```

Завдання №5 Epic 5 Task 7 - Lab# programming: Algotester Lab 6

```
1  #include <unordered_set>
2  #include <algorithm>
3  #include <iostream>
4  #include <vector>
5  #include <string>
6  using namespace std;
7  struct kord {
8      int x;
9      int y;
10 };
11
12 bool isInDesk(int x, int y) {
13     return x >= 0 && x < 8 && y >= 0 && y < 8;
14 }
15 string checkAttack(int x, int y, char desk[8][8]) {
16     if (desk[x][y] != '0') return "X";
17     unordered_set<char> attackers;
18
19     for (int i = 1; i < 8; i++) {
20         if (isInDesk(x - i, y + i) && (desk[x - i][y + i] == 'B' || desk[x - i][y + i] == 'Q')) attackers.insert(desk[x - i][y + i]);
21         if (isInDesk(x - i, y - i) && (desk[x - i][y - i] == 'B' || desk[x - i][y - i] == 'Q')) attackers.insert(desk[x - i][y - i]);
22         if (isInDesk(x + i, y - i) && (desk[x + i][y - i] == 'B' || desk[x + i][y - i] == 'Q')) attackers.insert(desk[x + i][y - i]);
23         if (isInDesk(x + i, y + i) && (desk[x + i][y + i] == 'B' || desk[x + i][y + i] == 'Q')) attackers.insert(desk[x + i][y + i]);
24     }
25
26     int kingMoves[8][2] = {{1, 1}, {1, -1}, {-1, -1}, {-1, 1}, {0, 1}, {0, -1}, {1, 0}, {-1, 0}};
27     for (auto &move : kingMoves) {
28         if (isInDesk(x + move[0], y + move[1]) && desk[x + move[0]][y + move[1]] == 'K') {
29             attackers.insert('K');
30         }
31     }
32
33     int knightMoves[8][2] = {{-2, -1}, {-2, 1}, {-1, -2}, {-1, 2}, {1, -2}, {1, 2}, {2, -1}, {2, 1}};
34     for (auto &move : knightMoves) {
35         if (isInDesk(x + move[0], y + move[1]) && desk[x + move[0]][y + move[1]] == 'N') {
36             attackers.insert('N');
37         }
38     }
39
40     if (isInDesk(x - 1, y - 1) && desk[x - 1][y - 1] == 'P') attackers.insert('P');
41     if (isInDesk(x - 1, y + 1) && desk[x - 1][y + 1] == 'P') attackers.insert('P');
42
43     for (int i = 1; i < 8; i++) {
44         if (isInDesk(x - i, y) && (desk[x - i][y] == 'R' || desk[x - i][y] == 'Q')) attackers.insert(desk[x - i][y]);
45         if (isInDesk(x + i, y) && (desk[x + i][y] == 'R' || desk[x + i][y] == 'Q')) attackers.insert(desk[x + i][y]);
46         if (isInDesk(x, y - i) && (desk[x][y - i] == 'R' || desk[x][y - i] == 'Q')) attackers.insert(desk[x][y - i]);
47         if (isInDesk(x, y + i) && (desk[x][y + i] == 'R' || desk[x][y + i] == 'Q')) attackers.insert(desk[x][y + i]);
48     }
49
50     string res(attackers.begin(), attackers.end());
51     sort(res.begin(), res.end());
52     return res.empty() ? "0" : res;
53 }
54
55 }
```

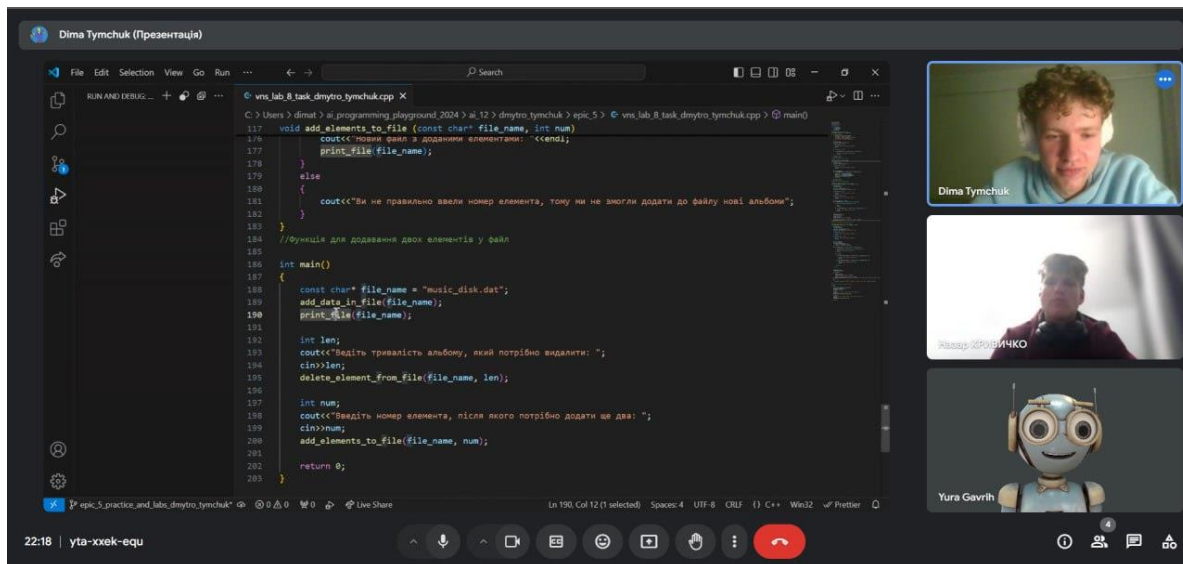
```
60 int main() {
61     char desk[8][8];
62     for (int i = 0; i < 8; i++) {
63         for (int j = 0; j < 8; j++) {
64             cin >> desk[i][j];
65         }
66     }
67     int n;
68     cin >> n;
69     vector<kord> requests(n);
70     for(int i = 0; i < n; i++){
71         cin >> requests[i].x >> requests[i].y;
72     }
73     for(int i = 0; i < n; i++){
74         cout << checkAttack(requests[i].x - 1, requests[i].y - 1, desk) << endl;
75     }
76
77     return 0;
78 }
79
```

Завдання №6 Epic 5 Task 8 - Practice# programming: Class Practice Task

```
1  #include<iostream>
2  #include<string>
3  #include<stdio.h>
4  using namespace std;
5
6  enum FileOpResult {Success, Failure};
7  FileOpResult write_to_file(const char *name, const char *content){
8      if (name == nullptr || strcmp(name, "") == 0 || !hasEnding(name, ".txt")) {
9          return FileOpResult::Failure;
10     }
11     FILE* fileStream;
12     fileStream = fopen(name, "w");
13     if (fileStream == nullptr){
14         cerr << "Не вдалося створити файл";
15         return FileOpResult::Failure;
16     }
17     fputs(content, fileStream);
18     fclose(fileStream);
19     return FileOpResult::Success;
20 }
21 FileOpResult copy_file(const char *file_from, const char *file_to){
22     if (name == nullptr || strcmp(name, "") == 0 || !hasEnding(name, ".txt")) {
23         return FileOpResult::Failure;
24     }
25     FILE* fileStream1;
26     fileStream1 = fopen(file_from, "r");
27     if (fileStream1 == nullptr){
28         cerr << "Не вдалося створити файл";
29         return FileOpResult::Failure;
30     }
31     FILE* fileStream2;
32     fileStream2 = fopen(file_to, "w");
33     if (fileStream2 == nullptr){
34         cerr << "Не вдалося створити файл";
35         fclose(fileStream1);
36         return FileOpResult::Failure;
37     }
38     const int size= 64;
39     char arr[size];
40     while(fgets(arr, size, fileStream1)){
41         fputs(arr, fileStream2);
42     }
43     fclose(fileStream1);
44     fclose(fileStream2);
45 }
46
47
48 int main() {
49     string name, content;
50     cin >> name;
51     cin >> content;
52     const char* namec = name.c_str();
53     const char* contentc = content.c_str();
54     string file_from, file_to;
55     cin >> file_from;
56     cin >> file_to;
57     const char* file_fromc = file_from.c_str();
58     const char* file_toc = file_to.c_str();
59     // write_to_file(namec, contentc);
60     copy_file(file_fromc, file_toc);
61     return 0;
62 }
```

Завдання №7 Task 9 - Practice# programming: Self Practice Task

Робота в команді



Висновок:

Під час виконання лабораторної роботи, я навчився використовувати різні методи для роботи з файлами, вивчив кілька бібліотечних функцій для швидшої роботи з масивами