

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

Виконала:

Студентка групи ІІІ-12

Смачило Іванна

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:

Дослідити основи роботи з файлами у мові програмування C++, зокрема розглянути та освоїти принципи роботи з текстовими та бінарними файлами. Опанувати операції введення та виведення символів і рядкових змінних у файл, а також ознайомитися зі стандартною бібліотекою C++ для роботи з файлами. Навчитися створювати власні бібліотеки та використовувати їх у проектах, організовуючи код для повторного використання та покращення його структури.

Теоретичні відомості

1. Поняття файла: http://cpp.dp.ua/ponyattya-fajla/#google_vignette
2. Базовий файловий ввід і вивід:
<https://acode.com.ua/urok-220-bazovyy-fajlovyj-vvid-i-vyvid/>
3. Файли в C++:
https://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/47_C++/index.html
4. Опрацювання рядків:
https://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/73_C++/index.html
<https://acode.com.ua/urok-208-ryadkovi-klasy-std-string-i-std-wstring/>

5. Символьний тип даних char:

<https://acode.com.ua/urok-38-symvolnyj-typ-danyh-char/>

6. Літерали і магичні числа:

<https://acode.com.ua/urok-39-litaly-i-magichni-chysla/>

7. Читання і запис бінарних файлів:

<https://studfile.net/preview/3904495/page:8/>

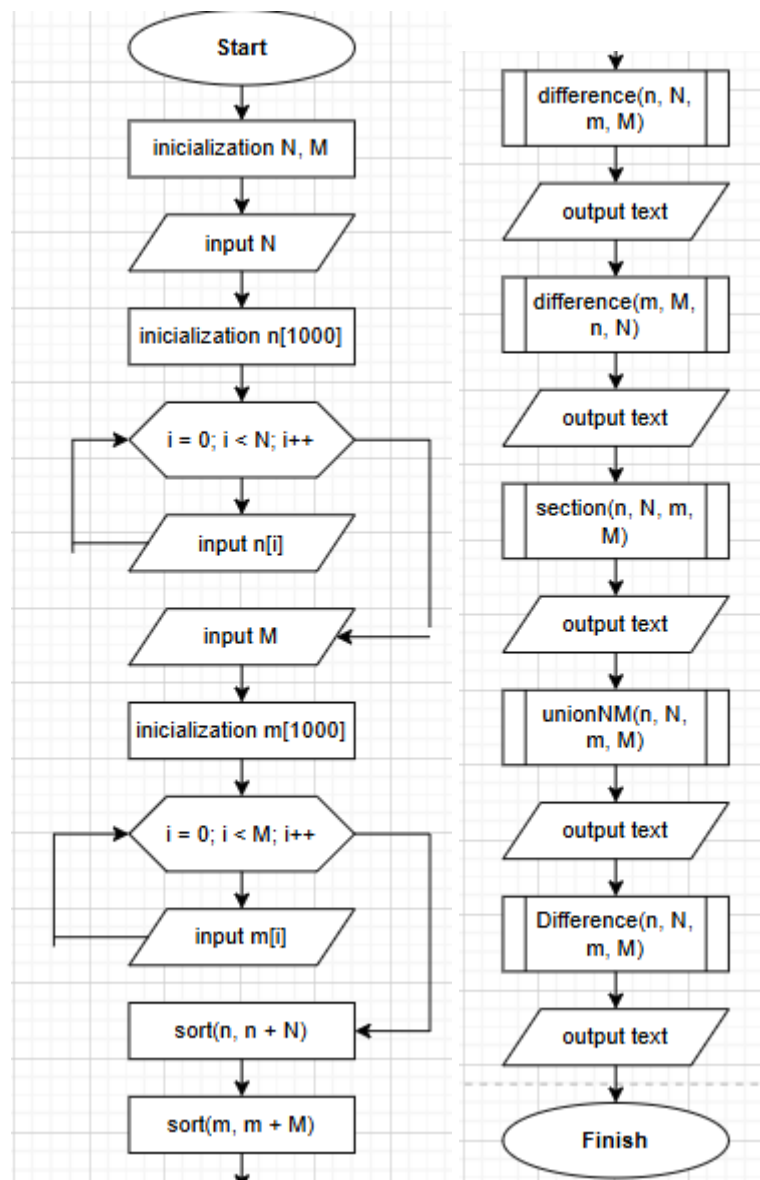
8. Fsream: <https://uk.wikipedia.org/wiki/Fstream>

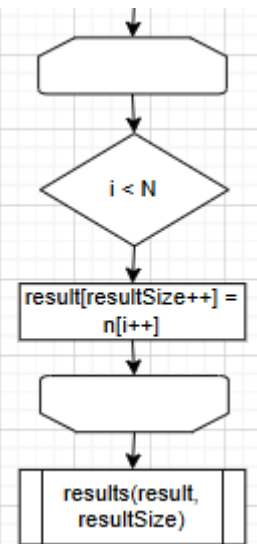
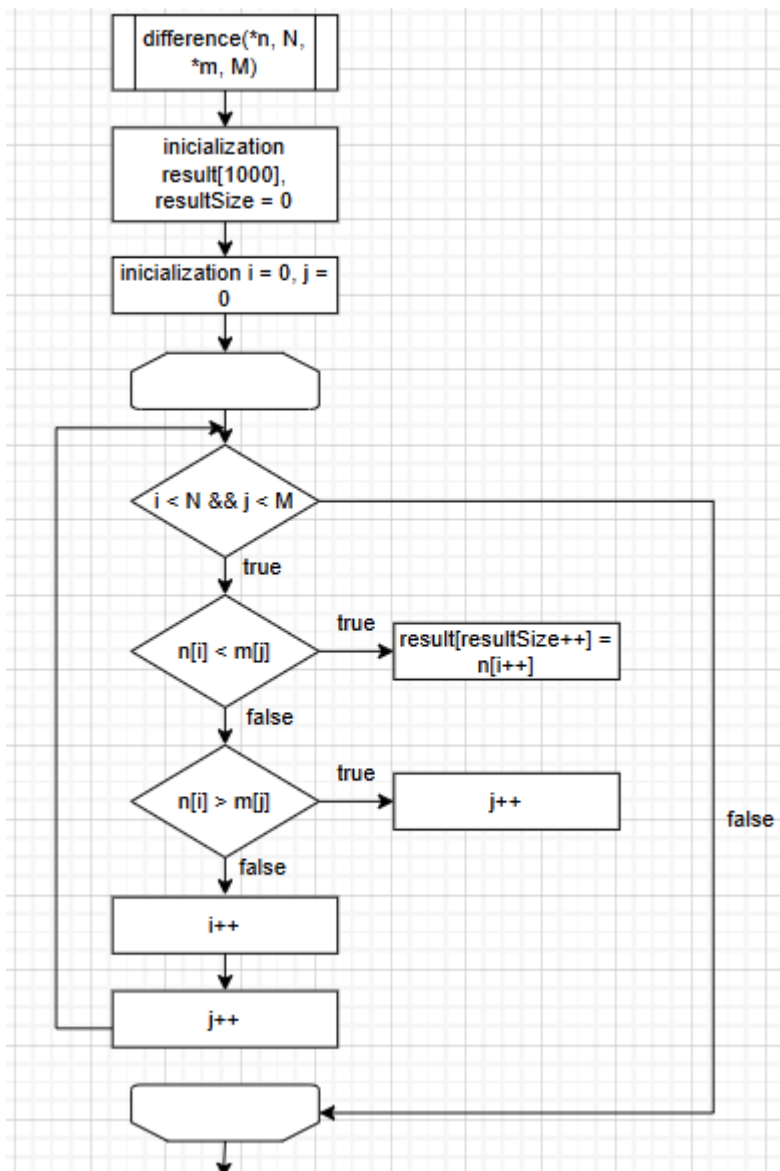
9. Створення і використання бібліотек:

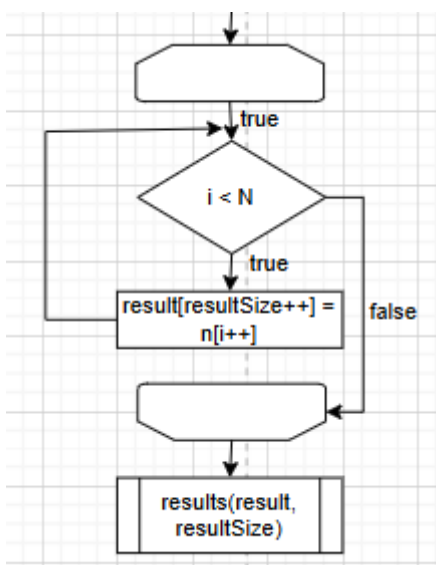
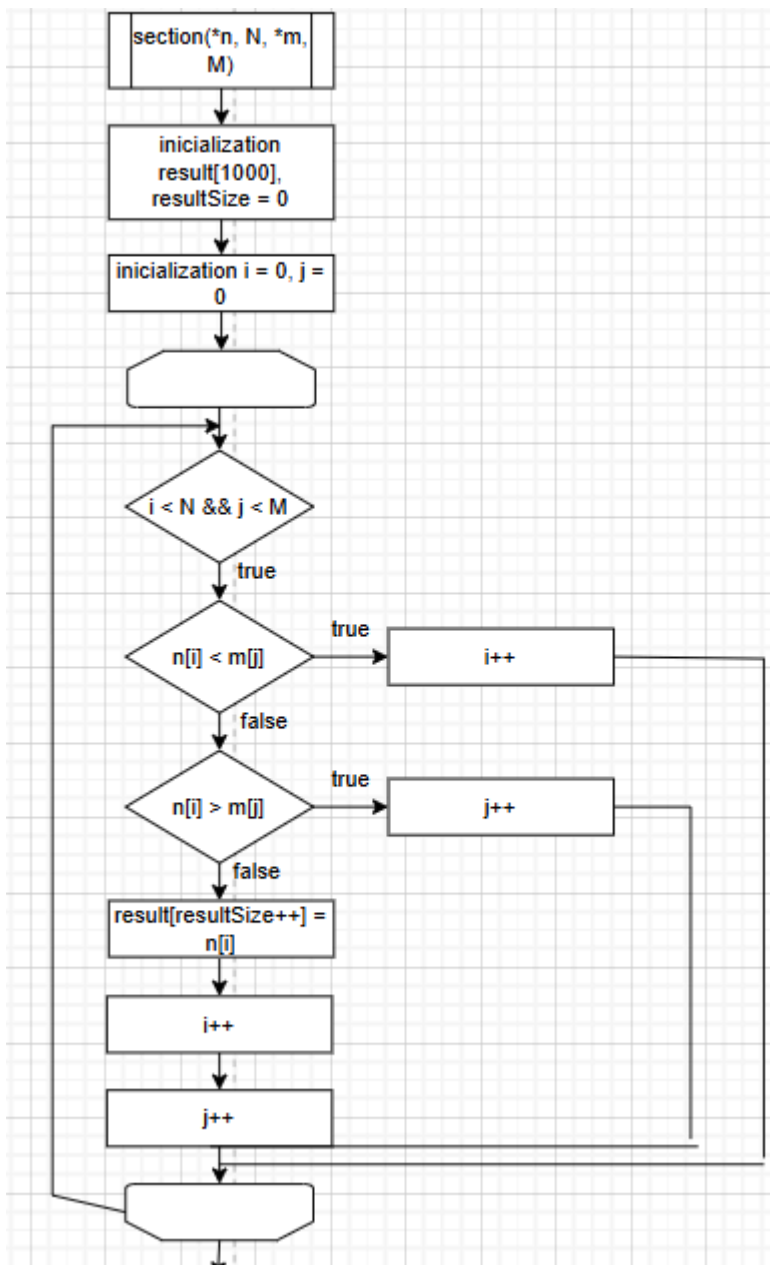
https://www.kievoit.ippo.kubg.edu.ua/kievoit/2016/66_C++/index.html

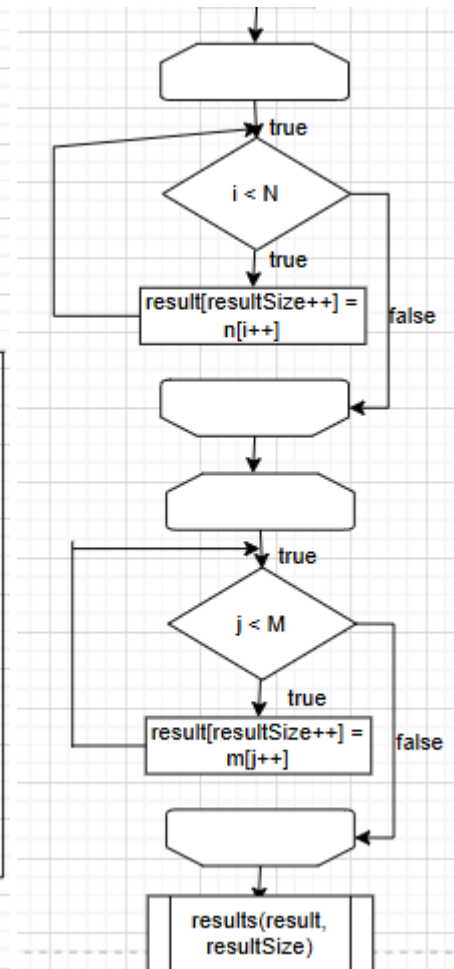
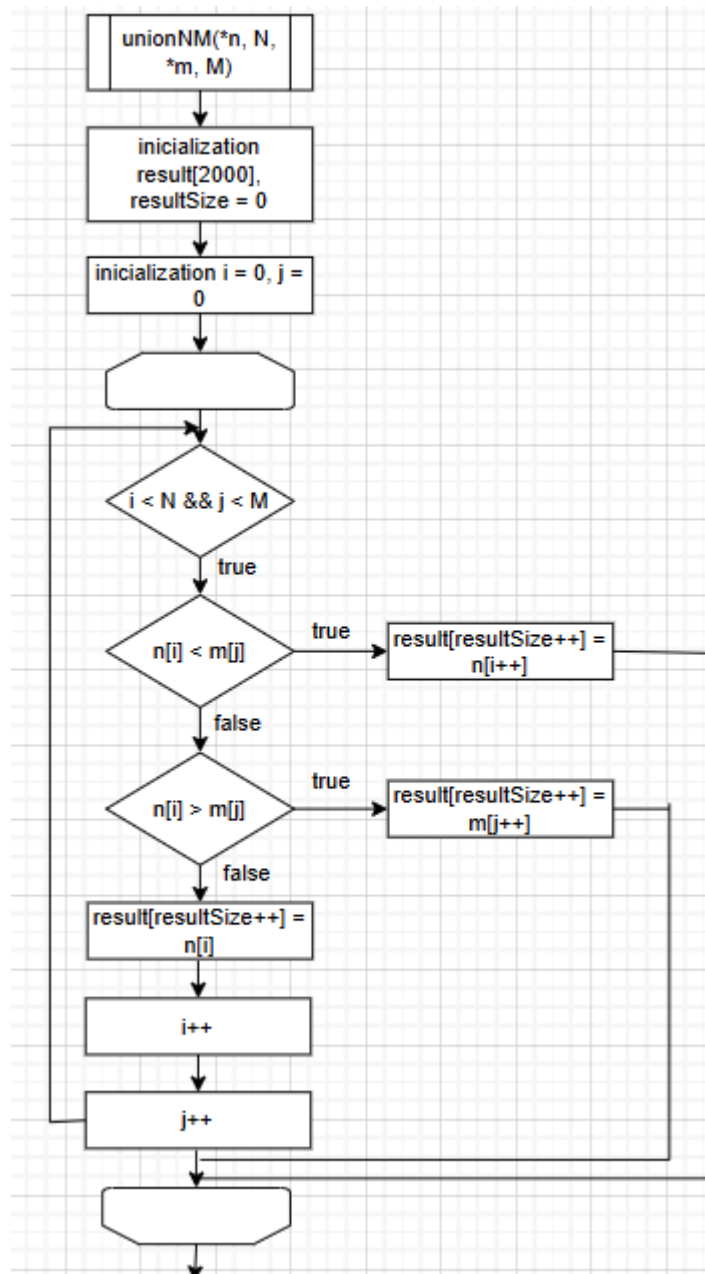
Виконання роботи

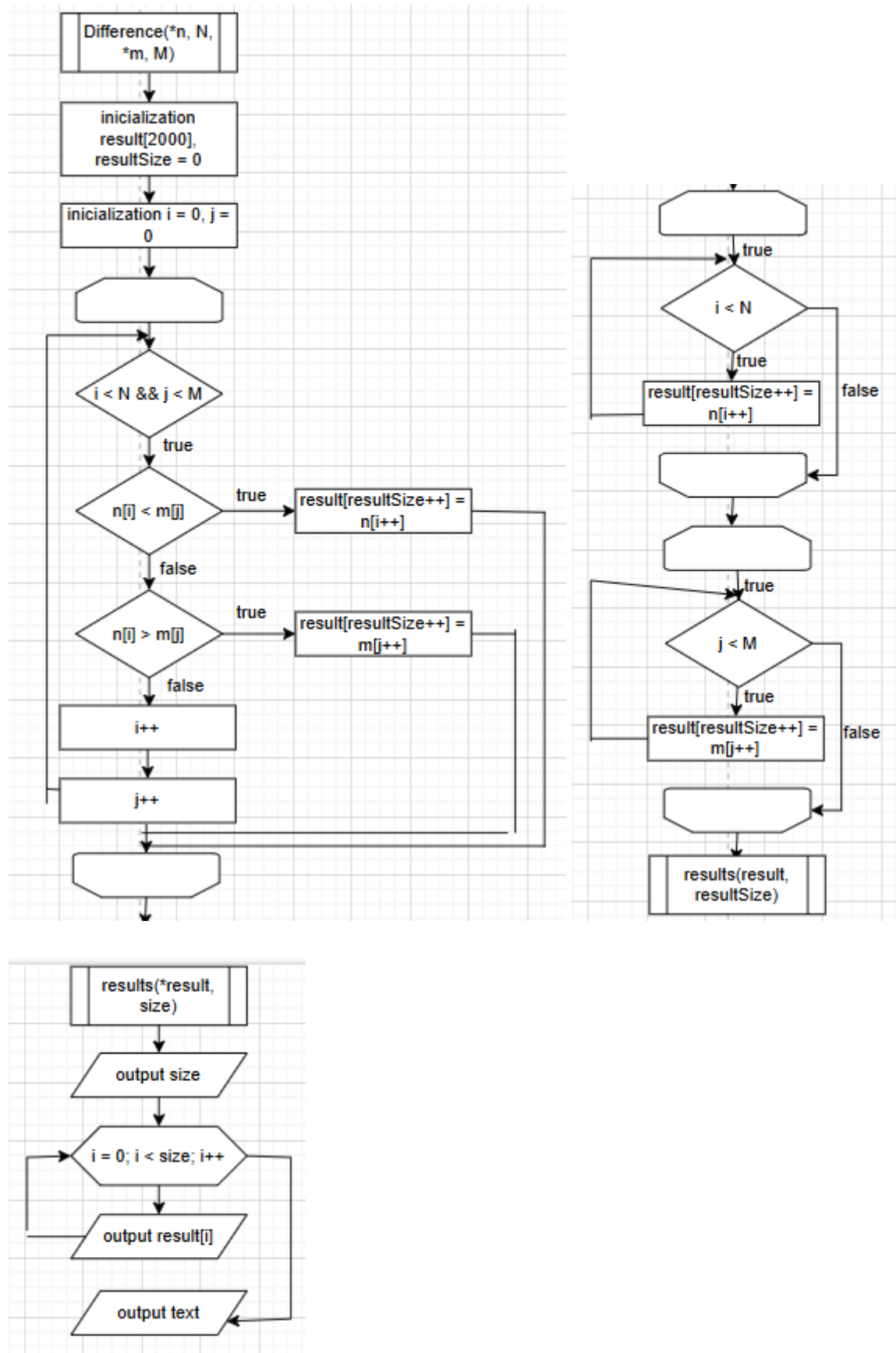
Task 2 - Requirements management (understand tasks) and design activities (draw flow diagram for task Algotester Lab 4 variant 1)











Task 3 - Lab# programming: VNS Lab 6(30 xB)

Завдання: Всі слова рядка, які починаються з букви, відсортувати за абеткою

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <ctype.h>
4
5  void SortWords(char str[])
6  {
7      char *words[256];
8      int count = 0;
9      char *token;
10
11     token = strtok(str, " .");
12     while (token != NULL)
13     {
14         if (isalpha(token[0]))
15         {
16             words[count++] = token;
17         }
18         token = strtok(NULL, " .");
19     }
20
21     for (int i = 0; i < count - 1; i++)
22     {
23         for (int j = i + 1; j < count; j++)
24         {
25             if (strcmp(words[i], words[j]) > 0)
26             {
27                 char *temp = words[i];
28                 words[i] = words[j];
29                 words[j] = temp;
30             }
31         }
32     }
33
34     printf("Words in alphabetical order: \n");
35     for (int i = 0; i < count; i++)
36     {
37         printf("%s\n", words[i]);
38     }
39 }
40
41 int main()
42 {
43     char s[256] = "banana apple strawberry fruits.";
44
45     SortWords(s);
46
47     return 0;
48 }
```

```
Words in alphabetical order:  
apple  
banana  
fruits  
strawberry
```

Task 4 - Lab# programming: VNS Lab 8(1 год 20 хв)

Завдання: Структура "Фільм":

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

Знищити всі елементи, у яких вартість перевищує задану, додати елемент на початок файлу

```
1  #include <iostream>  
2  #include <cstdio>  
3  #include <vector>  
4  #include <cstring>  
5  
6  using namespace std;  
7  
8  struct Movie  
9  {  
10     char title[100];  
11     char director[50];  
12     int year;  
13     double price;  
14 };  
15  
16 const char *filename = "movies.bin";  
17  
18 void createFile()  
19 {  
20     FILE *file = fopen(filename, "wb");  
21     if (!file)  
22     {  
23         cerr << "Помилка відкриття файлу для запису." << "\n";  
24         return;  
25     }  
26  
27     int n;  
28     cout << "Введіть кількість фільмів: ";  
29     cin >> n;  
30  
31     for (int i = 0; i < n; ++i)  
32     {  
33         Movie movie;  
34         cout << "Назва фільму: ";  
35         cin.ignore();  
36         cin.getline(movie.title, 100);
```

```

37     cout << "Режисер: ";
38     cin.getline(movie.director, 50);
39     cout << "Рік випуску: ";
40     cin >> movie.year;
41     cout << "Ціна: ";
42     cin >> movie.price;
43
44     fwrite(&movie, sizeof(Movie), 1, file);
45 }
46
47 fclose(file);
48 }
49
50 void printFile()
51 {
52     FILE *file = fopen(filename, "rb");
53     if (!file)
54     {
55         cerr << "Помилка відкриття файлу для читання!" << "\n";
56         return;
57     }
58
59     Movie movie;
60     bool isEmpty = true;
61
62     while (fread(&movie, sizeof(Movie), 1, file))
63     {
64         isEmpty = false;
65         cout << "Назва: " << movie.title
66              << ", Режисер: " << movie.director
67              << ", Рік випуску: " << movie.year
68              << ", ціна: " << movie.price << "\n";
69     }

```

```

70
71     if (isEmpty)
72     {
73         cout << "Файл порожній." << "\n";
74     }
75
76     fclose(file);
77 }
78
79 void deleteExpensiveMovies(double maxPrice)
80 {
81     FILE *file = fopen(filename, "rb");
82     if (!file)
83     {
84         cerr << "Помилка відкриття файлу для читання." << "\n";
85         return;
86     }
87
88     vector<Movie> movies;
89     Movie movie;
90
91     while (fread(&movie, sizeof(Movie), 1, file))
92     {
93         if (movie.price <= maxPrice)
94         {
95             movies.push_back(movie);
96         }
97     }
98
99     fclose(file);
100
101     file = fopen(filename, "wb");
102     if (!file)
103     {
104         cerr << "Помилка відкриття файлу для запису." << "\n";
105     }

```

```

105         return;
106     }
107
108     for (const auto &m : movies)
109     {
110         fwrite(&m, sizeof(Movie), 1, file);
111     }
112
113     fclose(file);
114 }
115
116 void addMovieToStart()
117 {
118     Movie newMovie;
119     cout << "Додавання нового фільму:" << "\n";
120     cout << "Назва: ";
121     cin.ignore();
122     cin.getline(newMovie.title, 100);
123     cout << "Режисер: ";
124     cin.getline(newMovie.director, 50);
125     cout << "Рік випуску: ";
126     cin >> newMovie.year;
127     cout << "Ціна: ";
128     cin >> newMovie.price;
129
130     FILE *file = fopen(filename, "rb");
131     if (!file)
132     {
133         cerr << "Помилка відкриття файлу для читання!" << "\n";
134         return;
135     }
136
137     vector<Movie> movies;
138     Movie movie;

```

```

139
140     while (fread(&movie, sizeof(Movie), 1, file))
141     {
142         movies.push_back(movie);
143     }
144
145     fclose(file);
146
147     file = fopen(filename, "wb");
148     if (!file)
149     {
150         cerr << "Помилка відкриття файлу для запису." << "\n";
151         return;
152     }
153
154     fwrite(&newMovie, sizeof(Movie), 1, file);
155
156     for (const auto &m : movies)
157     {
158         fwrite(&m, sizeof(Movie), 1, file);
159     }
160
161     fclose(file);
162 }

```

```

163
164 int main()
165 {
166     createFile();
167
168     cout << "\n" << "Вміст файлу після створення:" << "\n";
169     printFile();
170
171     double maxPrice;
172     cout << "\n" << "Максимальна ціна для збереження фільмів: ";
173     cin >> maxPrice;
174     deleteExpensiveMovies(maxPrice);
175
176     cout << "\n" << "Вміст файлу після видалення дорогих фільмів:" << "\n";
177     printFile();
178
179     addMovieToStart();
180
181     cout << "\n" << "Вміст файлу після додавання нового фільму:" << "\n";
182     printFile();
183
184     return 0;
185 }

```

Введіть кількість фільмів: 3

Назва фільму: Dune

Режисер: Denis Villeneuve

Рік випуску: 2021

Ціна: 100

Назва фільму: Interstellar

Режисер: Christopher Nolan

Рік випуску: 2010

Ціна: 120.5

Назва фільму: The Matrix

Режисер: Lana Wachowski

Рік випуску: 1999

Ціна: 99

Вміст файлу після створення:

Назва: Dune, Режисер: Denis Villeneuve, Рік випуску: 2021, Ціна: 100

Назва: Interstellar, Режисер: Christopher Nolan, Рік випуску: 2010, Ціна: 120.5

Назва: The Matrix, Режисер: Lana Wachowski, Рік випуску: 1999, Ціна: 99

Максимальна ціна для збереження фільмів: 100

Вміст файлу після видалення дорогих фільмів:

Назва: Dune, Режисер: Denis Villeneuve, Рік випуску: 2021, Ціна: 100

Назва: The Matrix, Режисер: Lana Wachowski, Рік випуску: 1999, Ціна: 99

Додавання нового фільму:

Назва: Mamma Mia!

Режисер: Phyllida Lloyd

Рік випуску: 2008

Ціна: 99.9

Вміст файлу після додавання нового фільму:

Назва: Mamma Mia!, Режисер: Phyllida Lloyd, Рік випуску: 2008, Ціна: 99.9

Назва: Dune, Режисер: Denis Villeneuve, Рік випуску: 2021, Ціна: 100

Назва: The Matrix, Режисер: Lana Wachowski, Рік випуску: 1999, Ціна: 99

Task 5 - Lab# programming: VNS Lab 9(1 год)

Завдання:

- 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.

```
1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <vector>
5
6  using namespace std;
7
8  int countWordsInLine(const string& line)
9  {
10     int wordCount = 0;
11     bool inWord = false;
12
13     for (char c : line)
14     {
15         if (isspace(c))
16         {
17             if (inWord)
18             {
19                 ++wordCount;
20                 inWord = false;
21             }
22         }
23         else
24         {
25             inWord = true;
26         }
27     }
28
29     if (inWord)
30     {
31         ++wordCount;
32     }
33
34     return wordCount;
35 }
```

```

37 int main()
38 {
39     ifstream fileF1("F1.txt");
40     ofstream fileF2("F2.txt");
41
42     if (!fileF1.is_open() || !fileF2.is_open())
43     {
44         cerr << "Помилка відкриття файлу!" << "\n";
45         return 1;
46     }
47
48     int N1, N2;
49     cout << "Введіть N1 та N2: ";
50     cin >> N1 >> N2;
51
52     string line;
53     int lineNumber = 0;
54     vector<string> linesToWrite;
55
56     while (getline(fileF1, line))
57     {
58         ++lineNumber;
59
60         if (lineNumber >= N1 && lineNumber <= N2 && !line.empty() && line[0] == 'A')
61         {
62             linesToWrite.push_back(line);
63         }
64
65         if (lineNumber > N2 + 2)
66         {
67             linesToWrite.push_back(line);
68         }
69     }
70

```

```

70
71     for (const auto& l : linesToWrite)
72     {
73         fileF2 << l << "\n";
74     }
75
76     fileF1.close();
77     fileF2.close();
78
79     if (!linesToWrite.empty())
80     {
81         const string& lastLine = linesToWrite.back();
82         int wordCount = countWordsInLine(lastLine);
83         cout << "Кількість слів в останньому рядку F2: " << wordCount << "\n";
84     }
85     else
86     {
87         cout << "Файл F2 порожній." << "\n";
88     }
89
90     return 0;
91 }

```

Введіть N1 та N2: 2 6

Кількість слів в останньому рядку F2: 4

```

C: > Epic_5 > F1.txt
1 Anne is reading a book.
2 Petro is working in the garden.
3 Andrew is playig tennis.
4 Misha is riding a bicycle.
5 Anastasia is doing her homework.
6 Artur is watching YouTube.
7 Nikita is coddig.

```

```

C: > Epic_5 > F2.txt
1 Andrew is playig tennis.
2 Anastasia is doing her homework.
3 Artur is watching YouTube.
4 |

```

Task 6 - Lab# programming: Algotester Lab 4(2 год 20 хв)

```

1  #include <stdio.h>
2  #include <algorithm>
3  #include <iostream>
4
5  using namespace std;
6
7  int main()
8  {
9      int N;
10     scanf("%d", &N);
11
12     int *array = new int[N];
13
14     for (int i = 0; i < N; i++)
15     {
16         scanf("%d", &array[i]);
17     }
18
19     int* zero = partition(array, array + N, [](int i) { return i % 3 == 0; });
20     int* one = partition(zero, array + N, [](int i) { return i % 3 == 1; });
21
22     sort(array, zero);
23     sort(zero, one, [](int a, int b)
24     {
25         return a > b;});
26     sort(one, array + N);

```

```

27
28     int *end = unique(array, array + N);
29     int new_size = distance(array, end);
30
31     printf("%d\n", new_size);
32     for (int i = 0; i < new_size; i++)
33     {
34         printf("%d ", array[i]);
35     }
36
37     delete[] array;
38
39     return 0;
40 }

```


Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
a few seconds ago	C++ 23	Accepted	0.003	1.211	View

```

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

```

```

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

```

```

37
38     while (i < N)
39     {
40         result[resultSize++] = n[i++];
41     }
42
43     results(result, resultSize);
44 }
45
46 void section(const int *n, int N, const int *m, int M)
47 {
48     int result[1000], resultSize = 0;
49
50     int i = 0, j = 0;
51     while (i < N && j < M)
52     {
53         if (n[i] < m[j])
54         {
55             i++;
56         }
57         else if (n[i] > m[j])
58         {
59             j++;
60         }
61         else
62         {
63             result[resultSize++] = n[i];
64             i++;
65             j++;
66         }
67     }
68
69     results(result, resultSize);
70 }
71

```

```

72 void unionMM(const int *n, int N, const int *m, int M)
73 {
74     int result[2000], resultSize = 0;
75
76     int i = 0, j = 0;
77     while (i < N && j < M)
78     {
79         if (n[i] < m[j])
80         {
81             result[resultSize++] = n[i++];
82         }
83         else if (n[i] > m[j])
84         {
85             result[resultSize++] = m[j++];
86         }
87         else
88         {
89             result[resultSize++] = n[i];
90             i++;
91             j++;
92         }
93     }
94
95     while (i < N)
96     {
97         result[resultSize++] = n[i++];
98     }
99
100    while (j < M)
101    {
102        result[resultSize++] = m[j++];
103    }
104
105    results(result, resultSize);
106 }

```

```

108 void Difference(const int *n, int N, const int *m, int M)
109 {
110     int result[2000], resultSize = 0;
111
112     int i = 0, j = 0;
113     while (i < N && j < M)
114     {
115         if (n[i] < m[j])
116         {
117             result[resultSize++] = n[i++];
118         }
119         else if (n[i] > m[j])
120         {
121             result[resultSize++] = m[j++];
122         }
123         else
124         {
125             i++;
126             j++;
127         }
128     }
129
130     while (i < N)
131     {
132         result[resultSize++] = n[i++];
133     }
134
135     while (j < M)
136     {
137         result[resultSize++] = m[j++];
138     }
139
140     results(result, resultSize);
141 }

```

```

143 int main() {
144     int N, M;
145
146     cin >> N;
147     int n[1000];
148     for (int i = 0; i < N; i++)
149     {
150         cin >> n[i];
151     }
152
153     cin >> M;
154     int m[1000];
155     for (int i = 0; i < M; i++)
156     {
157         cin >> m[i];
158     }
159
160     sort(n, n + N);
161     sort(m, m + M);
162
163     difference(n, N, m, M);
164     cout << "\n";
165
166     difference(m, M, n, N);
167     cout << "\n";
168
169     section(n, N, m, M);
170     cout << "\n";
171
172     unionNM(n, N, m, M);
173     cout << "\n";
174
175     Difference(n, N, m, M);
176     cout << "\n";
177
178     return 0;
179 }

```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
a few seconds ago	C++ 23	Accepted	0.003	1.211	View

```

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)

```

1  #include <iostream>
2  #include <cctype>
3  #include <vector>
4  #include <fstream>
5  #include <string>
6  #include <fstream>
7  #include <set>
8
9  using namespace std;
10
11 vector<int> values(vector<vector<int>> &v1, int x, int y, int n)
12 {
13     vector<int> result;
14     if(v1[x][y] != 0)
15     {
16         result.push_back(v1[x][y]);
17         return result;
18     }
19
20     set<int> forbidden;
21
22     for(int i = 0; i < n; i++)
23     {
24         if(v1[i][y] != 0)
25         {
26             forbidden.insert(v1[i][y]);
27         }
28     }
29
30     for(int i = 0; i < n; i++)
31     {
32         if(v1[x][i] != 0)
33         {
34             forbidden.insert(v1[x][i]);
35         }
36     }

```

```

37
38     for(int i = 1; i <= n; i++)
39     {
40         if(forbidden.find(i) == forbidden.end())
41         {
42             result.push_back(i);
43         }
44     }
45
46     return result;
47 }
48
49 int main()
50 {
51     int N, Q, x, y = 0;
52     string row;
53     cin >> N;
54     if(N > 9 || N < 1)
55     {
56         return 0;
57     }
58
59     vector<vector<int>> v1(N, vector<int>(N));
60     for(int i = 0; i < N; i++)
61     {
62         cin >> row;
63         if(row.length() != N)
64         {
65             return 0;
66         }
67         for(int j = 0; j < N; j++)
68         {
69             v1[i][j] = row[j] - 48;
70             if(v1[i][j] > 9 || v1[i][j] < 0)
71             {

```

```
72         return 0;
73     }
74 }
75
76
77     cin >> Q;
78
79     if(Q > 1000 || Q < 1)
80     {
81         return 0;
82     }
83
84     vector<pair<int, int>> position;
85     for(int i = 0; i < Q; i++)
86     {
87         cin >> x >> y;
88         x--;
89         y--;
90         position.push_back({x, y});
91     }
92
93     for(int i = 0; i < Q; i++)
94     {
95         vector<int> possible = values(v1, position[i].first, position[i].second, N);
96         int s = possible.size();
97         cout << s << endl;
98         for(int j = 0; j < s; j++)
99         {
100             cout << possible[j] << " ";
101         }
102         cout << "\n";
103     }
104 }
```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
a few seconds ago	C++ 23	Accepted	0.003	1.211	View

```
3
000
100
003
3
1 1
2 3
2 1
2
2 3
1
2
1
1
```

Task 8 - Practice# programming: Class Practice Task(2 год)

```

1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  enum FileOpResult{Success, Failure};
6
7  FileOpResult write_to_file(const char *name, char *content)
8  {
9      ifstream fin;
10
11      string line;
12
13      string StringContent(content);
14
15      fin.open(name);
16
17      if(!fin.is_open())
18      {
19          cerr << "Error: unable to open the file";
20          return Failure;
21      }
22
23      if(getline(fin, line))
24      {
25          if(line != StringContent)
26          {
27              cerr << "Error: input is not equal to file content";
28              return Failure;
29          }
30      }
31      else
32      {
33          cerr << "Error: file is clear";
34          return Failure;
35      }

```

```

37      fin.close();
38
39      return Success;
40  }
41
42  FileOpResult copy_file(const char *file_from, const char *file_to)
43  {
44      ofstream fto;
45      ifstream ffrom;
46
47      vector<string> lines_from;
48      vector<string> lines_to;
49
50      string line;
51
52      fto.open(file_to);
53      ffrom.open(file_from);
54      if(!fto.is_open())
55      {
56          cerr << "Error: unable to open the file 'to'" << "\n";
57          return Failure;
58      }
59
60      if(!ffrom.is_open())
61      {
62          cerr << "Error: unable to open the file 'from'" << "\n";
63          return Failure;
64      }
65
66      while(getline(ffrom, line))
67      {
68          fto << line;
69      }
70
71      fto.close();

```

```

72     ffrom.close();
73
74     ifstream ftoCheck;
75     ifstream ffromCheck;
76
77     ftoCheck.open(file_to);
78     ffromCheck.open(file_from);
79
80     while(getline(ffromCheck, line))
81     {
82         lines_from.push_back(line);
83     }
84
85     while(getline(ftoCheck, line))
86     {
87         lines_to.push_back(line);
88     }
89
90     if(lines_from.size() == 0)
91     {
92         cerr << "Error: file 'from' is clear" << "\n";
93         return Failure;
94     }
95
96     if(lines_to.size() == 0)
97     {
98         cerr << "Error: file 'to' is clear" << "\n";
99         return Failure;
100    }
101
102    for(int i = 0; i < lines_from.size(); i++)
103    {
104        if(lines_from[i] != lines_to[i])
105        {
106            cerr << "Error: contents of the two files are not equal" << "\n";

```

```

107            return Failure;
108        }
109    }
110
111    ftoCheck.close();
112    ffromCheck.close();
113
114    return Success;
115 }
116
117 int main()
118 {
119     ofstream fout;
120
121     const char *FileName1 = "C:/Users/ivank/ai_programming_playground_2024/ai_12/ivanna_smachylo/epic_5/file_from.txt";
122     const char *FileName2 = "C:/Users/ivank/ai_programming_playground_2024/ai_12/ivanna_smachylo/epic_5/file_to.txt";
123
124     char content[256];
125
126     fout.open(FileName1);
127
128     if(!fout.is_open())
129     {
130         cout << Failure << "\n";
131         cerr << "Error: unable to open the file";
132         return 1;
133     }
134
135     cin.getline(content, 256);
136
137     fout << content;

```



```

138
139     fout.close();
140
141     cout << "Execution status of the task 1 : " << write_to_file(FileName1, content) << "\n";
142     cout << "Execution status of the task 2 : " << copy_file(FileName1, FileName2);
143
144     return 0;
145 }

```

```

Hello world!
Execution status of the task 1 : 0
Execution status of the task 2 : 0

```

```

C: > Users > ivank > ai_programming_playground_2024 > ai_12 > ivanna_smachylo > epic_5 > file_from.txt
1 Hello world!

```

```

C: > Users > ivank > ai_programming_playground_2024 > ai_12 > ivanna_smachylo > epic_5 > file_to.txt
1 Hello world!

```

Task 9 - Practice# programming: Self Practice Task

Стипендія

```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main()
7  {
8      int n;
9      cin >> n;
10
11      vector<int> grades(n);
12      bool fail = false;
13      bool excellent = true;
14
15      for (int i = 0; i < n; ++i)
16      {
17          cin >> grades[i];
18          if (grades[i] < 51)
19          {
20              fail = true;
21          }
22          if (grades[i] < 90)
23          {
24              excellent = false;
25          }
26      }

```

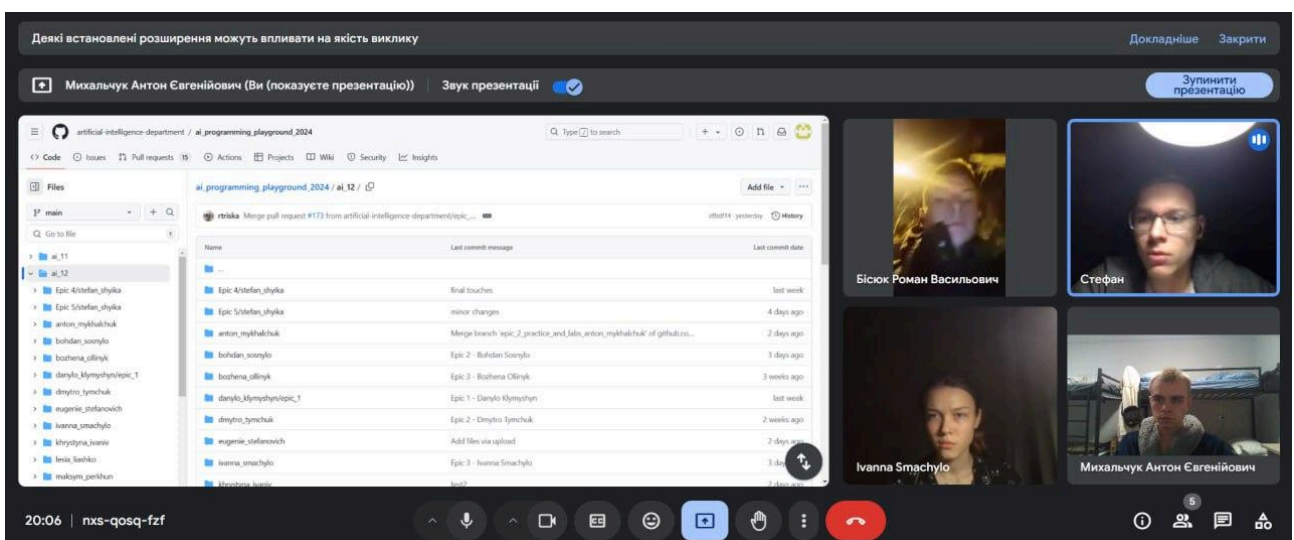
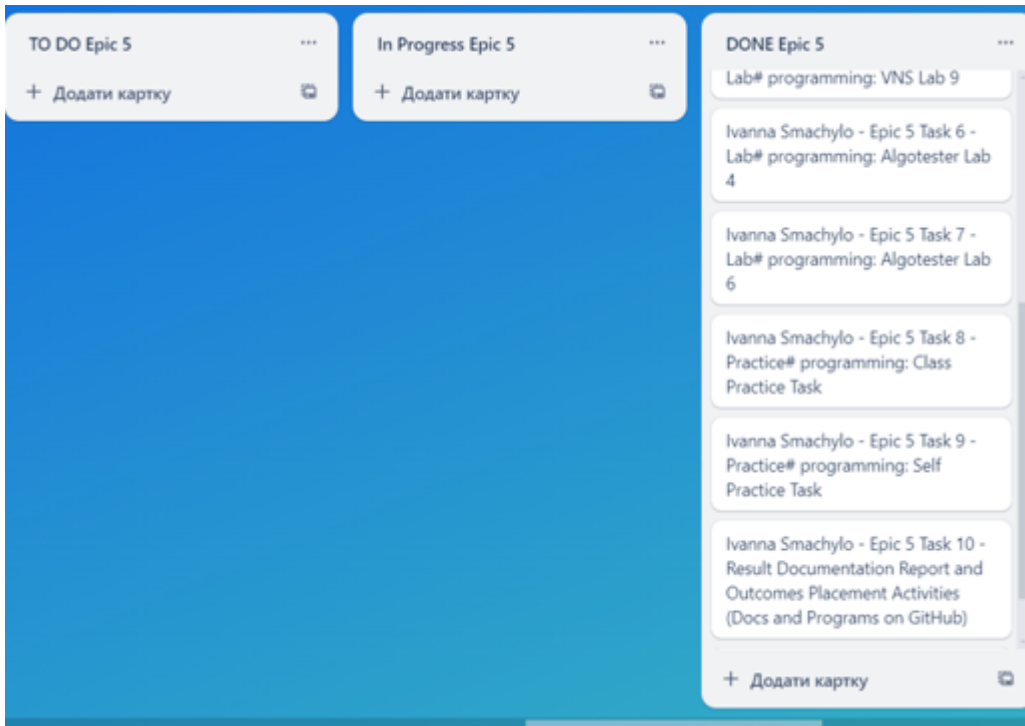
```
27
28     if (fail)
29     {
30         cout << "Zabud pro stypendiiu" << "\n";
31     }
32     else if (excellent)
33     {
34         cout << "Pidvyshchena" << "\n";
35     }
36     else
37     {
38         cout << "Zvychaina" << "\n";
39     }
40
41     return 0;
42 }
```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
a few seconds ago	C++ 23	Accepted	0.003	1.211	View

```
4
4 7 47 74
Zabud pro stypendiiu
```

Зустрічі з командою

Зустрічались двічі для обговорення задач, поставлених в п'ятому епіку. Створили нову дошку в Trello й бачили прогрес одне одного:



Висновок: в ході роботи над даним епіком я навчилась використовувати на практиці нові знання, такі як поняття файла, базовий файловий ввід і вивід, файли в C++, опрацювання рядків, символний тип даних char, Літерали і

магічні числа, читання і запис бінарних файлів, створення і використання бібліотек.