



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

Перетворити рядок таким чином, щоб букви кожного слова в ньому були відсортовані за зростанням.

```
1  #include<iostream>
2  #include<string>
3  using namespace std;
4  void bubble_sort(string& s)
5  {
6      int change_code;
7      char change_symbol;
8
9      for (int i = 0; i < s.length() - 1 ; i++)
10     {
11         for (int j = 0; j < s.length() - i - 1; j++)
12         {
13             if ((int)s[j] > (int)s[j + 1])
14             {
15                 change_symbol = s[j];
16                 s[j] = s[j + 1];
17                 s[j + 1] = change_symbol;
18             }
19         }
20     }
21 }
22
23 int main()
24 {
25     string s;
26     cin>>s;
27     int arr_of_code[1000];
28     for (int i = 0; i < s.length(); i++)
29     {
30         arr_of_code[i] = (int)s[i];
31     }
32     bubble_sort(s);
33
34     cout<<s<<endl;
35     return 0;
36 }
```

Тестування

```
dadadadfad
aaaadddddf
PS C:\Users\dimat> █
```

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

Структура "Музичний диск":

- назва;
- автор;
- тривалість;
- ціна.

Знищити перший елемент із заданою тривалістю, додати 2 елементи після елемента із заданим номером.

```
1  #include<iostream>
2  #include<stdio.h>
3  using namespace std;
4
5  struct Music_disk
6  {
7      char name[100];
8      char author[100];
9      int duration;
10     double price;
11 };
12 //Функція для запису структури у файл
13 void add_data_in_file(const char* file_name)
14 {
15     Music_disk disks[]={
16         {"Meteora", "Linkin park", 40, 30.0},
17         {"Nevermind", "Nirvana", 35, 25.5},
18         {"Придумано в черзі", "Паліндром", 35, 42.25},
19         {"8 роздумів", "ОТОУ", 28, 25.50},
20         {"False light", "White ward", 50, 33.30}
21     };
22
23     FILE* music_file;
24     music_file = fopen(file_name, "wb");
25
26     if (music_file == nullptr)
27     {
28         cerr<<"Не вдалося відкрити файл для запису!";
29         exit(0);
30     }
31
32     for (int i = 0; i < 5; i++)
33     {
34         if (fwrite(&disks[i], sizeof(Music_disk), 1, music_file) != 1)
35         {
36             cerr<<"Не вдалося записати дані у файл!";
37             exit(0);
38         }
39     }
40     fclose(music_file);
41 }
42 //Функція для запису структури у файл
```

```

44 //Функція для виведення даних з файлу
45 void print_file(const char* file_name)
46 {
47     FILE* music_file;
48     music_file = fopen(file_name, "rb");
49
50     if (music_file == nullptr)
51     {
52         cerr<<"Не вдалося відкрити файл для читання!";
53         exit(0);
54     }
55
56     Music_disk new_disk;
57     while (fread(&new_disk, sizeof(Music_disk), 1, music_file) == 1)
58     {
59         cout<<"Назва: "<<new_disk.name<<endl;
60         cout<<"Автор: "<<new_disk.author<<endl;
61         cout<<"Довжина: "<<new_disk.duration<<endl;
62         cout<<"Ціна: "<<new_disk.price<<endl;
63         cout<<endl;
64     }
65
66     fclose(music_file);
67 }
68 //Функція для виведення даних з файлу

```

```

0 //Функція для видалення елемента з заданою довжиною
1 void delete_element_from_file(const char* file_name, int len)
2 {
3     FILE* music_file;
4     FILE* new_music_file;
5
6     music_file = fopen(file_name, "rb");
7     new_music_file = fopen("new_music_disk.dat", "wb");
8
9     if (music_file == nullptr)
10    {
11        cerr<<"Не вдалося відкрити файл для читання!";
12        exit(0);
13    }
14    if (new_music_file == nullptr)
15    {
16        cerr<<"Не вдалося відкрити файл для запису!";
17        exit(0);
18    }
19
20    Music_disk new_disk; int flag = 0;
21    while (fread(&new_disk, sizeof(Music_disk), 1, music_file))
22    {
23        if ((flag == 0) && (new_disk.duration == len))
24        {
25            flag = 1;
26            continue;
27        }
28        else if (fwrite(&new_disk, sizeof(Music_disk), 1, new_music_file) != 1)
29        {
30            cerr<<"Не вдалося записати дані у файл!";
31            exit(0);
32        }
33    }
34
35    fclose(music_file);
36    fclose(new_music_file);
37
38    remove(file_name);
39    rename("new_music_disk.dat", file_name);
40
41    cout<<"Файл після видалення диску з заданою довжиною: "<<endl;
42    print_file(file_name);
43 }
44 //Функція для видалення елемента з заданою довжиною

```

```

116 //Функція для додавання двох елементів у файл
117 void add_elements_to_file (const char* file_name, int num)
118 {
119     FILE* music_file;
120     FILE* new_music_file;
121
122     music_file = fopen(file_name, "rb");
123     new_music_file = fopen("new_music_disks.dat", "wb");
124
125     if (music_file == nullptr)
126     {
127         cerr<<"Не вдалося відкрити файл для читання!";
128         exit(0);
129     }
130     if (new_music_file == nullptr)
131     {
132         cerr<<"Не вдалося відкрити файл для запису!";
133         exit(0);
134     }
135
136     int i = 1, flag = 0;
137     Music_disk new_disk;
138     Music_disk new_two_disks[] {
139         {"Encore", "Eminem", 45, 18.33},
140         {"Amnesiac", "Radiohead", 43, 25.75}
141     };
142     while (fread(&new_disk, sizeof(Music_disk), 1, music_file) == 1)
143     {
144         if (fwrite(&new_disk, sizeof(Music_disk), 1, new_music_file) != 1)
145         {
146             cerr<<"Не вдалося записати дані у файл!";
147             exit(0);
148         }
149
150         if (i == num)
151         {
152             if (fwrite(&new_two_disks[0], sizeof(Music_disk), 1, new_music_file) != 1)
153             {
154                 cerr<<"Не вдалося записати дані у файл!";
155                 exit(0);
156             }

```

```

157             if (fwrite(&new_two_disks[1], sizeof(Music_disk), 1, new_music_file) != 1)
158             {
159                 cerr<<"Не вдалося записати дані у файл!";
160                 exit(0);
161             }
162
163             flag = 1;
164         }
165         i++;
166     }
167
168     fclose(music_file);
169     fclose(new_music_file);
170
171     if (flag == 1)
172     {
173         remove(file_name);
174         rename("new_music_disks.dat", file_name);
175
176         cout<<"Новий файл з доданими елементами: "<<endl;
177         print_file(file_name);
178     }
179     else
180     {
181         cout<<"Ви не правильно ввели номер елемента, тому ми не змогли додати до файлу нові альбоми";
182     }
183 }
184 //Функція для додавання двох елементів у файл
185
186 int main()
187 {
188     const char* file_name = "music_disk.dat";
189     add_data_in_file(file_name);
190     print_file(file_name);
191
192     int len;
193     cout<<"Ведіть тривалість альбому, який потрібно видалити: ";
194     cin>>len;
195     delete_element_from_file(file_name, len);
196

```

```

    int num;
    cout<<"Введіть номер елемента, після якого потрібно додати ще два: ";
    cin>>num;
    add_elements_to_file(file_name, num);

    return 0;
}

```

Тестування

```
Назва: Meteora
Автор: Linkin park
Довжина: 40
Ціна: 30

Назва: Nevermind
Автор: Nirvana
Довжина: 35
Ціна: 25.5

Назва: Придуманo в черзі
Автор: Паліндром
Довжина: 35
Ціна: 42.25

Назва: 8 роздумів
Автор: ОТОУ
Довжина: 28
Ціна: 25.5

Назва: False light
Автор: White ward
Довжина: 50
Ціна: 33.3

Ведіть тривалість альбому, який потрібно видалити:
```

```
Ведіть тривалість альбому, який потрібно видалити: 35
Файл після видалення диску з заданою довжиною:
Назва: Meteora
Автор: Linkin park
Довжина: 40
Ціна: 30

Назва: Придуманo в черзі
Автор: Паліндром
Довжина: 35
Ціна: 42.25

Назва: 8 роздумів
Автор: ОТОУ
Довжина: 28
Ціна: 25.5

Назва: False light
Автор: White ward
Довжина: 50
Ціна: 33.3
```

```
Введіть номер елемента, після якого потрібно додати ще два: 1
Новий файл з доданими елементами:
Назва: Meteora
Автор: Linkin park
Довжина: 40
Ціна: 30

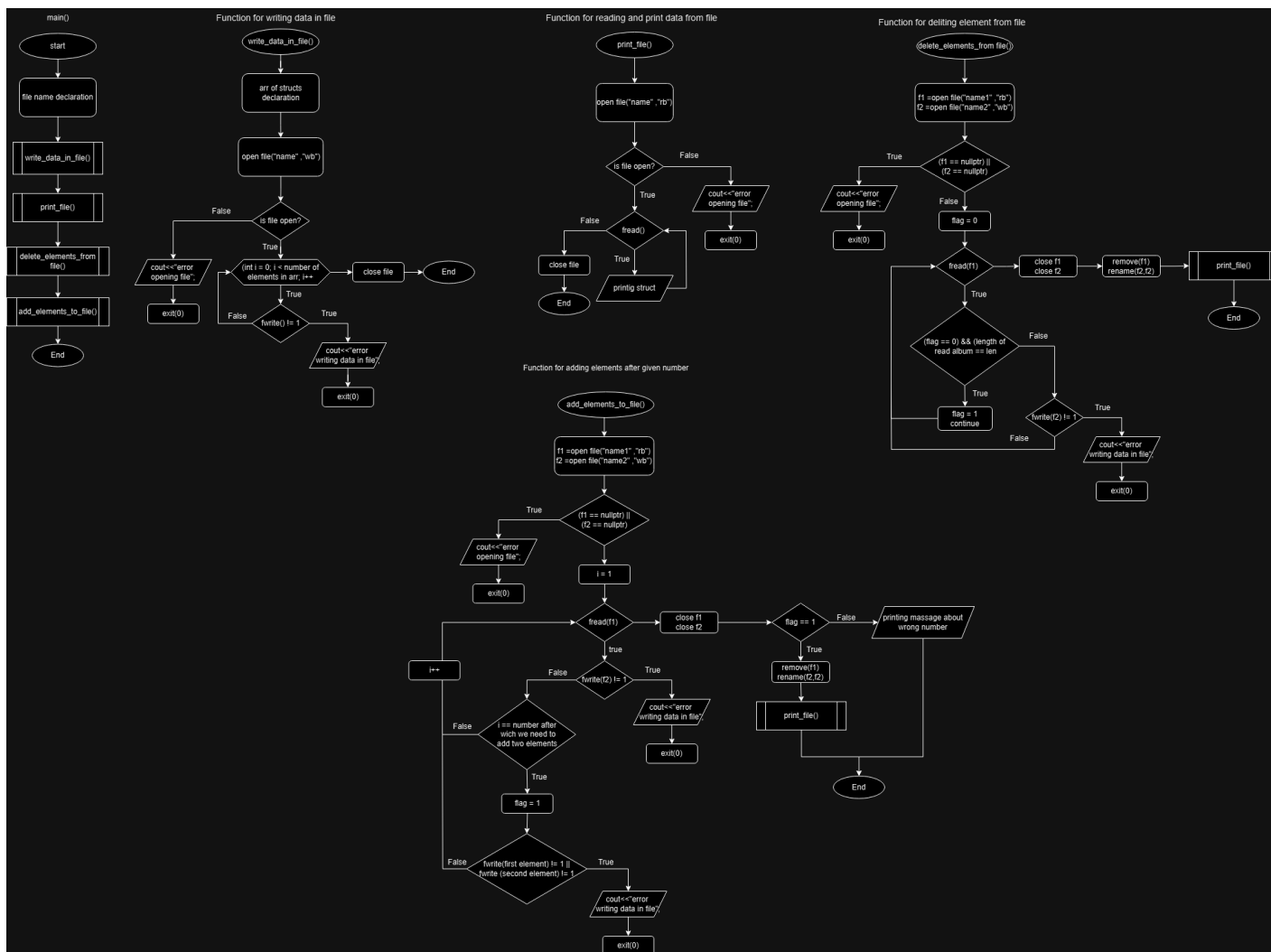
Назва: Encore
Автор: Eminem
Довжина: 45
Ціна: 18.33

Назва: Amnesiac
Автор: Radiohead
Довжина: 43
Ціна: 25.75

Назва: Придуманo в черзі
Автор: Паліндром
Довжина: 35
Ціна: 42.25

Назва: 8 роздумів
Автор: ОТОУ
Довжина: 28
Ціна: 25.5

Назва: False light
Автор: White ward
Довжина: 50
Ціна: 33.3
```



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

- 1) Скопіювати з файлу F1 у файл F2 всі рядки, крім того рядка, у якій найбільше голосних букв.
- 2) Надрукувати номер цього рядка


```

1  #include<iostream>
2  #include<string>
3  #include<cstdio>
4  using namespace std;
5
6  int main()
7  {
8      FILE* f1 = fopen("input.txt", "r");
9      FILE* f2 = fopen("output.txt", "w");
10
11      char s[256], max_s[256];
12      if(f1 == nullptr)
13      {
14          cerr<<"Не вдалося відкрити файл для читання!";
15          return 1;
16      }
17      if(f2 == nullptr)
18      {
19          cerr<<"Не вдалося відкрити файл для запису!";
20          return 1;
21      }
22
23      int max_num_of_vowels = 0, index = 0, max_index, num_of_vowels = 0;
24      while (fgets(s, sizeof(s), f1) != NULL)
25      {
26          num_of_vowels = 0;
27          index++;
28
29          for (int i = 0; i < sizeof(s); i++)
30          {
31              if ((s[i] == 'a') || (s[i] == 'e') || (s[i] == 'i') || (s[i] == 'o') || (s[i] == 'u') || (s[i] == 'y')) num_of_vowels++;
32          }
33
34          if (num_of_vowels > max_num_of_vowels)
35          {
36              max_num_of_vowels = num_of_vowels;
37              max_index = index;
38          }
39      }
40      fclose(f1);
41

```


```

index = 0;
f1 = fopen("input.txt", "r");
while (fgets(s, sizeof(s), f1) != NULL)
{
    index++;
    if (index != max_index)
    {
        fputs(s, f2);
    }
}
cout<<"Номер рядка з найбільшою кількістю голосних: "<<max_index<<endl;

fclose(f1);
fclose(f2);
return 0;
}

```

Тестування


 input: Блокнот

Файл Редагування Формат

```

dadadad
dsdad
frifk
adqwjidieqw
aaaaaaaaaaaaaaaaaaaa

```

 output: Блокнот

Файл Редагування Формат

```

dadadad
dsdad
frifk
adqwjidieqw

```

```

' '--stdout=Microsoft-MIEngine-Out-y02c2ron.haz' '--st
'--interpreter=mi'
Номер рядка з найбільшою кількістю голосних: 5
PS C:\Users\dimat>

```

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

Lab4v2 (version 1)

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

```
10 3
1 2 2 3 3 3 4 5 6 7
7
4 5 6 7 1 2 3
PS C:\Users\dimat>
```

Lab4v2 (version 2)

```
1  #include<iostream>
2  #include<vector>
3  using namespace std;
4
5  //швидке сортування
6  void quick_sort(vector<int>& arr, int first, int last)
7  {
8      int middle = arr[(first + last) / 2];
9      int left = first;
10     int right = last;
11     int change;
12
13     while (left <= right)
14     {
15         while (arr[right] > middle) right--;
16
17         while (arr[left] < middle) left++;
18
19         if (left <= right)
20         {
21             change = arr[left];
22             arr[left] = arr[right];
23             arr[right] = change;
24             left++;
25             right--;
26         }
27     }
28     if (right > first) quick_sort(arr, first, right);
29     if (left < last) quick_sort(arr, left, last);
30 }
31 //швидке сортування
32
```

```
//видалення однакових елементів
void erase_elements (vector<int>& arr)
{
    int i = 1;
    while (i < arr.size())
    {
        if (arr[i] == arr[i - 1])
        {
            arr.erase(arr.begin() + i);
        }
        else i++;
    }
}

//видалення однакових елементів

void arr_transformation(vector<int>& arr, vector<int>& new_arr, int k)
{
    int i = k;
    int size = 0;

    while (size < arr.size())
    {
        if (i < arr.size())
        {
            new_arr[size] = arr[i];
            i++;
            size++;
        }
        else i = 0;
    }
}
```

```
int main()
{
    int n, k;

    cin>>n>>k;
    vector<int> arr(n);
    for (int i = 0; i < n; i++)
    {
        cin>>arr[i];
    }

    quick_sort(arr, 0, arr.size() - 1);
    erase_elements(arr);
    vector<int> new_arr(arr.size());

    arr_transformation(arr, new_arr, k % arr.size());

    cout<<new_arr.size()<<endl;
    for (int i = 0; i < new_arr.size(); i++)
    {
        cout<<new_arr[i]<<" ";
    }

    return 0;
}
```

Lab4v3 (version 1)

```
1  #include<iostream>
2  #include<algorithm>
3  #include<vector>
4  using namespace std;
5
6  void print_arr (vector<int>& arr)
7  {
8      cout<<arr.size()<<endl;
9      for (int i = 0; i < arr.size(); i++)
10     {
11         cout<<arr[i]<<" ";
12     }
13 }
14 int remainder_0(int n)
15 {
16     return n % 3 == 0;
17 }
18 int remainder_1(int n)
19 {
20     return n % 3 == 1;
21 }
22
23 int main()
24 {
25     int n;
26     cin>>n;
27
28     vector<int> arr(n);
29     for (int i = 0; i < n; i++)
30     {
31         cin>>arr[i];
32     }
33 }
```

```
33
34 auto begin_1 = partition(arr.begin(), arr.end(), remainder_0);
35 sort(arr.begin(), begin_1);
36 auto index_1 = unique(arr.begin(), begin_1);
37 arr.erase(index_1, begin_1);
38
39 auto begin_2 = partition(index_1, arr.end(), remainder_1);
40 sort(index_1, begin_2);
41 auto index_2 = unique(index_1, begin_2);
42 arr.erase(index_2, begin_2);
43
44 sort(index_2, arr.end());
45 auto index_3 = unique(index_2, arr.end());
46 arr.erase(index_3, arr.end());
47
48 sort(index_1, index_2, [](int a, int b) {
49     return a > b;
50 });
51
52 print_arr(arr);
53 return 0;
54 }
```

```
--interpreter=mi'
10
1 33 4 8 6 5 2 7 5 0
9
0 6 33 7 4 1 2 5 8
PS C:\Users\dimat>
```

Lab4v3 (version 2)

```
1  #include<iostream>
2  #include<vector>
3  using namespace std;
4
5  //Виведення масиву
6  void print_arr (vector<int>& arr)
7  {
8      cout<<arr.size()<<endl;
9      for (int i = 0; i < arr.size(); i++)
10     {
11         cout<<arr[i]<<" ";
12     }
13 }
14 //Виведення масиву
15
16 //Перестановка елементів з остачею 0 на початок
17 int duration_0(vector<int>& arr)
18 {
19     int index = 0;
20     int change;
21     for (int i = 0; i < arr.size(); i++)
22     {
23         if (arr[i] % 3 == 0)
24         {
25             change = arr[index];
26             arr[index] = arr[i];
27             arr[i] = change;
28
29             index++;
30         }
31     }
32
33     return index;
34 }
35 //Перестановка елементів з остачею 0 на початок
```

```
//Перестановка елементів з остачею 1 в середину
int duration_1(vector<int>& arr, int begin_1)
{
    int index = begin_1;
    int change;
    for (int i = begin_1; i < arr.size(); i++)
    {
        if (arr[i] % 3 == 1)
        {
            change = arr[index];
            arr[index] = arr[i];
            arr[i] = change;
            index++;
        }
    }
    return index;
}
//Перестановка елементів з остачею 1 в середину
//Швидке сортування
void quick_sort(vector<int>& arr, int first, int last)
{
    int middle = arr[(first + last) / 2];
    int left = first;
    int right = last;
    int change;
    while (left <= right)
    {
        while (arr[right] > middle) right--;
        while (arr[left] < middle) left++;
        if (left <= right)
        {
            change = arr[left];
            arr[left] = arr[right];
            arr[right] = change;
            left++;
            right--;
        }
    }
}
```

```
80     }
81 }
82 if (right > first) quick_sort(arr, first, right);
83 if (left < last) quick_sort(arr, left, last);
84 }
85 //Швидке сортування
86
87 //Швидке сортування по спаданню
88 void quick_sort_fall(vector<int>& arr, int first, int last)
89 {
90     int middle = arr[(first + last) / 2];
91     int left = first;
92     int right = last;
93     int change;
94
95     while (left <= right)
96     {
97         while (arr[right] < middle) right--;
98         while (arr[left] > middle) left++;
99
100        if (left <= right)
101        {
102            change = arr[left];
103            arr[left] = arr[right];
104            arr[right] = change;
105            left++;
106            right--;
107        }
108    }
109    if (right > first) quick_sort_fall(arr, first, right);
110    if (left < last) quick_sort_fall(arr, left, last);
111 }
112 //Швидке сортування по спаданню
113
114 //Швидке сортування по спаданню
```

```
115 //Видалення однакових елементів
116 void erase_elements (vector<int>& arr, int& n)
117 {
118     int i = 1;
119     while (i < arr.size())
120     {
121         if (arr[i] == arr[i - 1])
122         {
123             arr.erase(arr.begin() + i);
124             n--;
125         }
126         else i++;
127     }
128 }
129 //Видалення однакових елементів
130
131 int main()
132 {
133     int n;
134     cin>>n;
135     vector<int> arr(n);
136     for (int i = 0; i < n; i++)
137     {
138         cin>>arr[i];
139     }
140
141     quick_sort(arr, 0, n - 1);
142     erase_elements(arr, n);
143
144     int begin_1 = duration_0(arr);
145     int begin_2 = duration_1(arr, begin_1);
146
147     quick_sort(arr, 0, begin_1 - 1);
148     quick_sort_fall(arr, begin_1, begin_2 - 1);
149     quick_sort(arr, begin_2, n - 1);
150
151     print_arr(arr);
152     return 0;
153 }
154 }
```

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

```
1 #include<iostream>
2 #include<string>
3 #include <cstdlib>
4 using namespace std;
5
6 int main()
7 {
8     string s;
9     int n, matrix[100][100];
10    cin>>n;
11    for (int i = 1; i <= n; i++)
12    {
13        cin>>s;
14        for (int j = 0; j < s.length(); j++)
15        {
16            matrix[i][j + 1] = (int)s[j] - 48;
17        }
18    }
19
20    int q, x[100], y[100];
21    cin>>q;
22    for (int i = 0; i < q; i++)
23    {
24        cin>>x[i]>>y[i];
25    }
26
27    for (int i = 0; i < q; i++)
28    {
29        int missing_elements[10] = {}, num_of_missing_elements = 0;
30
31        if (matrix[x[i]][y[i]] == 0)
32        {
33            for (int j = 1; j <= n; j++)
34            {
35                if ((matrix[x[i]][j] != 0) && (missing_elements[matrix[x[i]][j]] == 0))
36                {
37                    missing_elements[matrix[x[i]][j]] = 1;
38                    num_of_missing_elements++;
39                }
40            }
41        }
42    }
```

```
        if ((matrix[j][y[i]] != 0) && (missing_elements[matrix[j][y[i]]] == 0))
        {
            missing_elements[matrix[j][y[i]]] = 1;
            num_of_missing_elements++;
        }
    }

    if (n - num_of_missing_elements == 0) cout<<0;
    else
    {
        int m = 0;

        cout<<n - num_of_missing_elements<<endl;
        for (int j = 1; j <= n; j++)
        {
            if (missing_elements[j] == 0)
            {
                m++;
                if (m == n - num_of_missing_elements) cout<<j;
                else cout<<j<<" ";
            }
        }
        if (i < q - 1) cout<<endl<<endl;
    }
    else
    {
        printf("%d\n%d", 1, matrix[x[i]][y[i]]);
        if (i < q - 1) cout<<endl<<endl;
    }
}

return 0;
}
```

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

```
1
2
```

```
1
1
```

```
PS C:\Users\dimat>
```

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

```
1  #include<iostream>
2  #include<cstring>
3  using namespace std;
4
5  enum FileOpResult { Success, Failure};
6
7  FileOpResult add_to_file (const char* file_name_1, const char* content)
8  {
9      if (file_name_1 == nullptr)
10     {
11         return Failure;
12     }
13
14     FILE* f1;
15     f1 = fopen(file_name_1, "w");
16     if (f1 == nullptr)
17     {
18         return Failure;
19     }
20
21     if (fwrite(content, sizeof(char), strlen(content), f1) != strlen(content))
22     {
23         fclose(f1);
24         return Failure;
25     }
26
27     if (fclose(f1) != 0)
28     {
29         return Failure;
30     }
31
32     return Success;
33 }
34
```

```
FileOpResult copy_file(const char* file_from, const char* file_to)
{
    if (file_from == nullptr || file_to == nullptr)
    {
        return Failure;
    }

    FILE* f1;
    FILE* f2;
    f1 = fopen(file_from, "r");
    f2 = fopen(file_to, "w");

    if (file_from == nullptr || file_to == nullptr)
    {
        return Failure;
    }

    char c[256];

    while (fgets(c, sizeof(c), f1) != nullptr)
    {
        if (fputs(c, f2) == EOF)
        {
            {
                fclose(f1);
                fclose(f2);
                return Failure;
            }
        }
    }

    if (fclose(f1) != 0 || fclose(f2) != 0)
    {
        return Failure;
    }

    return Success;
}
```

```
int main()
{
    const char* file_name_1 = "file1.txt";
    const char* file_name_2 = "file2.txt";
    char content[256];

    cout<<"Введіть рядок, який треба вписати у файл: ";
    cin.getline(content, sizeof(content));

    if (add_to_file(file_name_1, content) == Failure)
    {
        cout<<"Не вдалося записати рядок до файлу"<<endl;
    }
    else cout<<"Рядок був успішно записаний до файлу"<<endl;

    if (copy_file(file_name_1, file_name_2) == Failure)
    {
        cout<<"Не вдалося скопіювати дані з першого файлу в другий"<<endl;
    }
    else cout<<"Дані з першого файлу були успішно скопійовані в другий"<<endl;

    return 0;
}
```

Тестування

```
Введіть рядок, який треба вписати у файл: akdoaidki dao oed as d;;dewkd; k
Рядок був успішно записаний до файлу
Дані з першого файлу були успішно скопійовані в другий
PS C:\Users\dimat>
```

file1: Блокнот

Файл Редагування Формат Вигляд Довідка
akdoaidki dao oed as d;;dewkd; k

file2: Блокнот

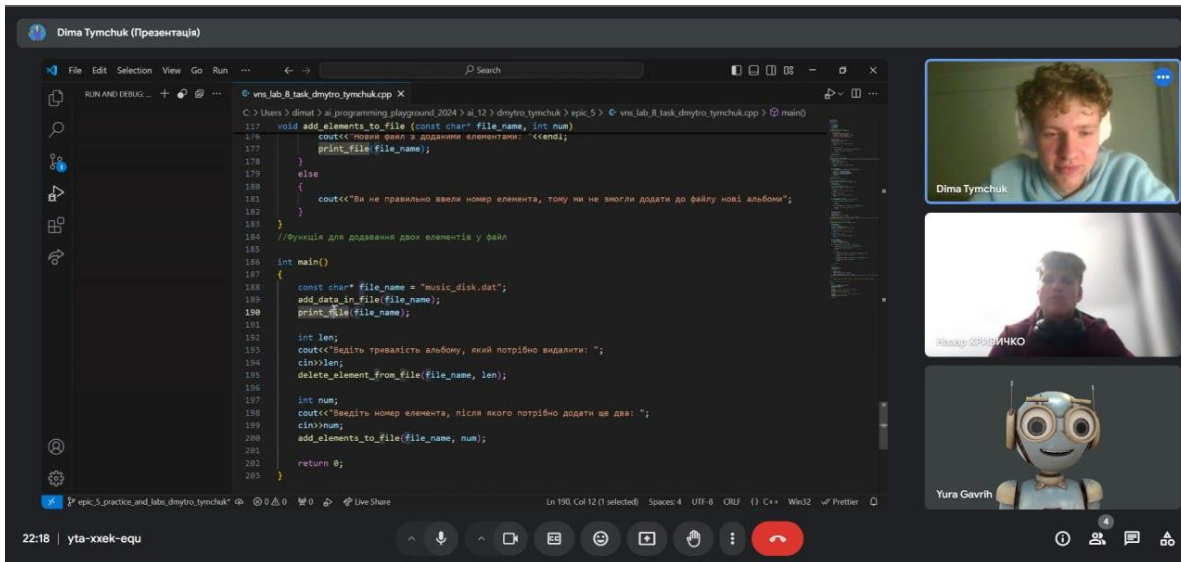
Файл Редагування Формат Вигляд Довідка
akdoaidki dao oed as d;;dewkd; k

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

```
1 //Ім'я судженого 434
2 #include<iostream>
3 #include<string>
4 using namespace std;
5
6
7 int main()
8 {
9     string s1, s2;
10    int arr[1000]{};
11
12    cin>>s1;
13    cin>>s2;
14
15    for (int i = 0; i < s1.length(); i++)
16    {
17        arr[(int)s1[i]]++;
18    }
19
20    int flag = 0;
21    for (int i = 0; i < s2.length(); i++)
22    {
23        if (arr[(int)s2[i]] == 0)
24        {
25            flag = 1;
26            break;
27        }
28        else arr[(int)s2[i]]--;
29    }
30
31    if (flag == 0) cout<<"YES"<<endl;
32    else cout<<"NO"<<endl;
33
34
35    return 0;
36 }
```

```
--stdout=Microsoft-MinGin
'--interpreter=mi'
noonaameeen
nononame
YES
PS C:\Users\dimat>
```

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