

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 ІІІ-12

Кутельмах Євген Петрович

Львів 2024

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

**Мета:** Навитись працювати з файлами у мові C++, зчитувати з/в них дані, отримати практичні навички з роботи з файлами, та рядковими змінними. Дізнатись, що таке Стандартна бібліотека, навчитись нею користуватись. Навчитись створювати власні бібліотеки та працювати з ними.

## 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/>

На цих уроках я прочитав теорії по роботі з файлами, це зайняло у мене 30 хв.

### 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/>

На цих уроках я навчився користуватися різними видами стрічок - як string, так і C-style, на це я витратив близько 40 хв.

### 3) Стандартна бібліотека:

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

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

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

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

На цих уроках я прочитав про стандартну бібліотеку шаблонів, зрозумів, що це дуже потужний інструмент та ознайомився з деякими класами цієї бібліотеки. Це зайняло у мене 40 хв.

## Виконання роботи:

### Task 3 - Lab# programming: VNS Lab 6

Умова: Задано рядок, що складається із символів. Символи поєднуються в слова. Слова одне від одного відокремлюються одним або декількома пробілами. Наприкінці тексту ставиться крапка. Текст містить не більше 255 символів. Виконати ввід рядка, використовуючи функцію gets(s) і здійснити обробку рядка у відповідності зі своїм варіантом.

Варіант 6: Перетворити рядок так, щоб всі букви в ньому були відсортовані за зростанням.

Загалом очікував написати дане завдання за 15 хв,десь так і вийшло.

Програма:

```
1  #include <cstring>
2  #include <iostream>
3  #include <algorithm>
4  using namespace std;
5
6  /*
7  Приклад:
8  The event starts at 9:00 AM on November 15, 2024, and will last for 3 hours. Please arrive by 8:45 AM. Contact us at 555-123-4567 for any que
9  */
10
11 /*
12 Перетворити рядок так, щоб всі букви в ньому були відсортовані за
13 зростанням.
14 */
15
16 bool isLetter(char c) {
17     if((c >= 65 && c <= 90) || (c >= 97 && c <= 122)) return true;
18     else return false;
19 }
20
21 void findLetters(char copy[]) {
22     for(int i = 0; i < strlen(copy); i++) {
23         if(!isLetter(copy[i]))
24             copy[i] = 24;
25     }
26 }
27
28 void sort(char* text) {
29     for(int i = 0; i < strlen(text); i++) {
30         if(text[i]==24) continue;
31         for(int j = i + 1; j < strlen(text); j++) {
32             if(text[j]==24) continue;
33             if(text[j] < text[i])
34                 swap(text[i], text[j]);
35         }
36     }
37 }
38
39 void changeStr(char* original, char* copy) {
40     for(int i = 0; i < strlen(original); i++) {
41         if(copy[i]!=24) {
42             original[i] = copy[i];
43         }
44     }
45 }
46
47 int main() {
48     char text[256];
49     gets(text);
50     char copy[256];
51     strcpy(copy, text);
52     findLetters(copy);
53     sort(copy);
54     changeStr(text, copy);
55     cout << text;
56     return 0;
57 }
```

Результат:

```
vns_lab_6_task_yevhen_kutelmakh.cpp: In function 'int main()':
vns_lab_6_task_yevhen_kutelmakh.cpp:49:9: warning: call to 'gets' declared with attribute warning: Using gets() is always unsafe - use fgets() instead [-Wattri
bute-warning]
49 |     gets(text);
    |     ~~~~~
The event starts at 9:00 AM on November 15, 2024, and will last for 3 hours. Please arrive by 8:45 AM. Contact us at 555-123-4567 for any questions. The regist
ration fee is $50 per person.
AAC MMNPТ Тааааа аа 9:00 аа ab bcdeeeee 15, 2024, eee eeee eeef ffg 3 hhhii. iiiill llmnnn nn 8:45 nn. nooooooo oo op 555-123-4567 pqr rrr rrrrrrrrss. sss ssssss
ttttt ttt tt $50 uuu vvvvvv.
```

## Task 4 - Lab# programming: VNS Lab 8

Умова: Сформувати двійковий файл із елементів, заданої у варіанті структури, роздрукувати його вмістиме, виконати знищення й додавання елементів у відповідності зі своїм варіантом, використовуючи для пошуку елементів що

знищуються чи додаються, функцію. Формування, друк, додавання й знищення елементів оформити у вигляді функцій. Передбачити повідомлення про помилки при відкритті файлу й виконанні операцій вводу/виводу.

Варіант 6: Структура "Школяр":

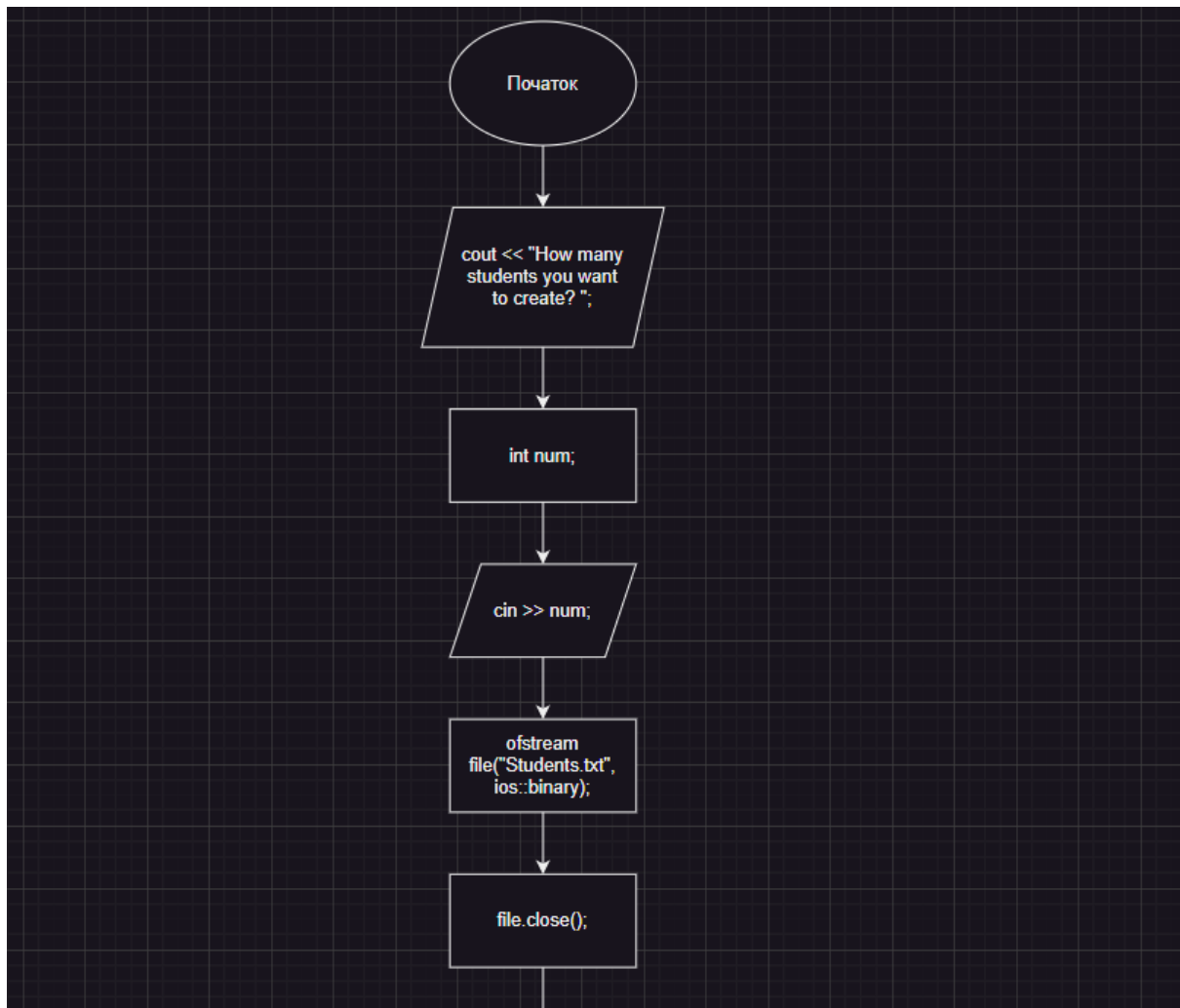
- прізвище, ім'я, по батькові;
- клас;
- номер телефону;
- оцінки по предметах (математика, фізика, українська мова, література).

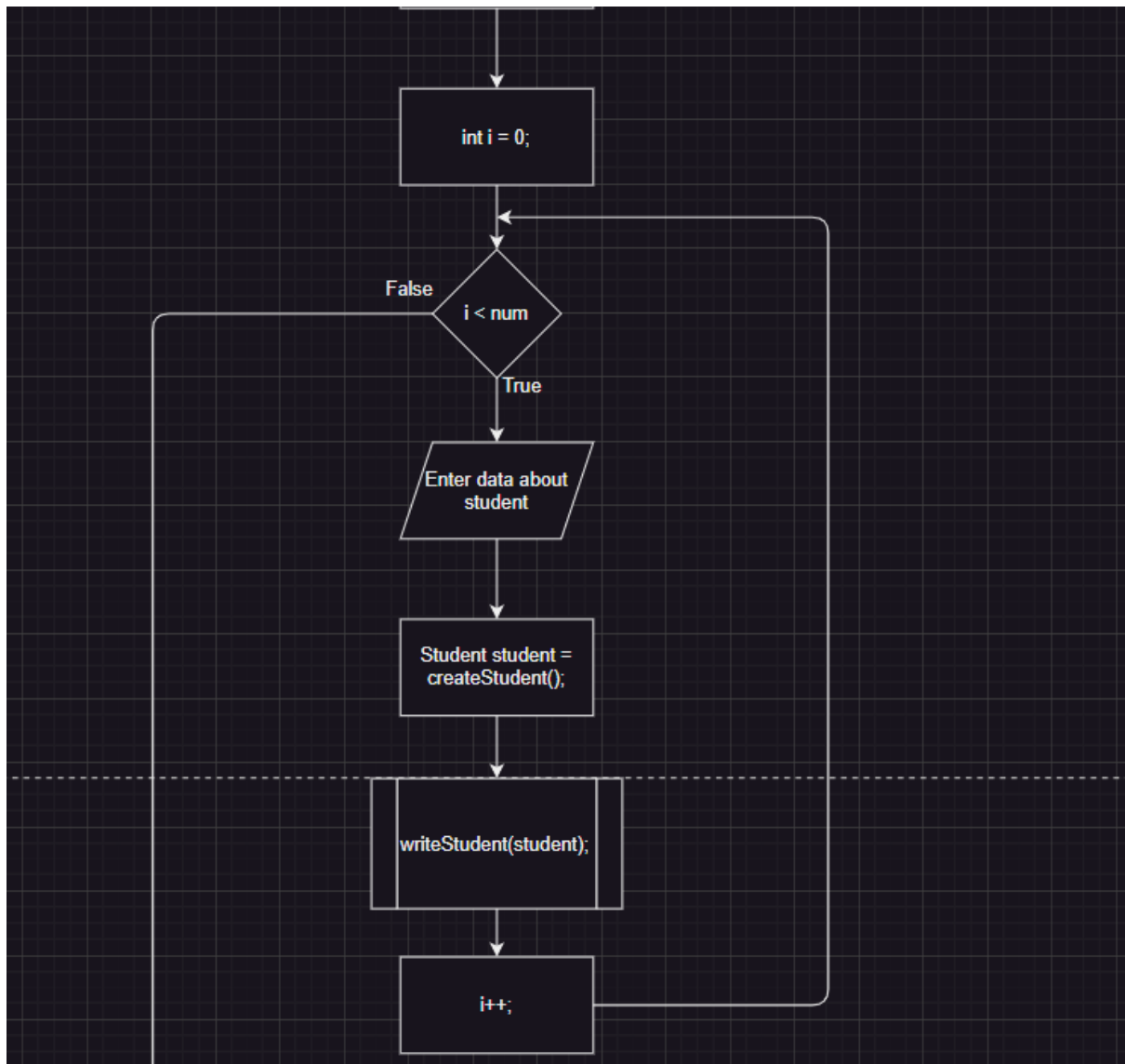
Знищити всі елементи, у яких є 2 хоча б з одного предмету, додати елемент у початок файлу.

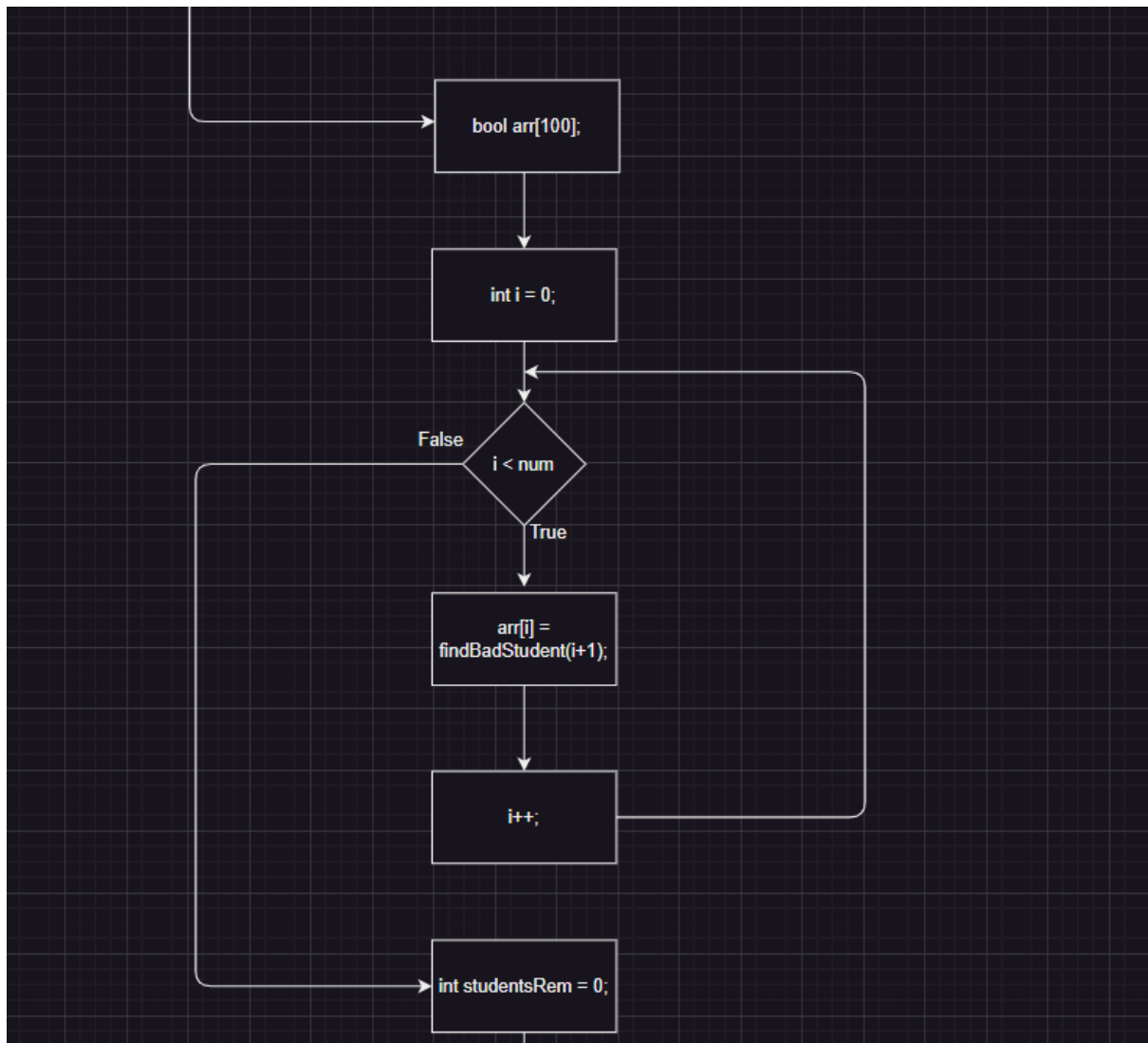
Спочатку я вважав, що напишу дане завдання за годину, проте воно виявилось складнішим ніж на перший погляд, тому на нього я витратив близько 3 годин.

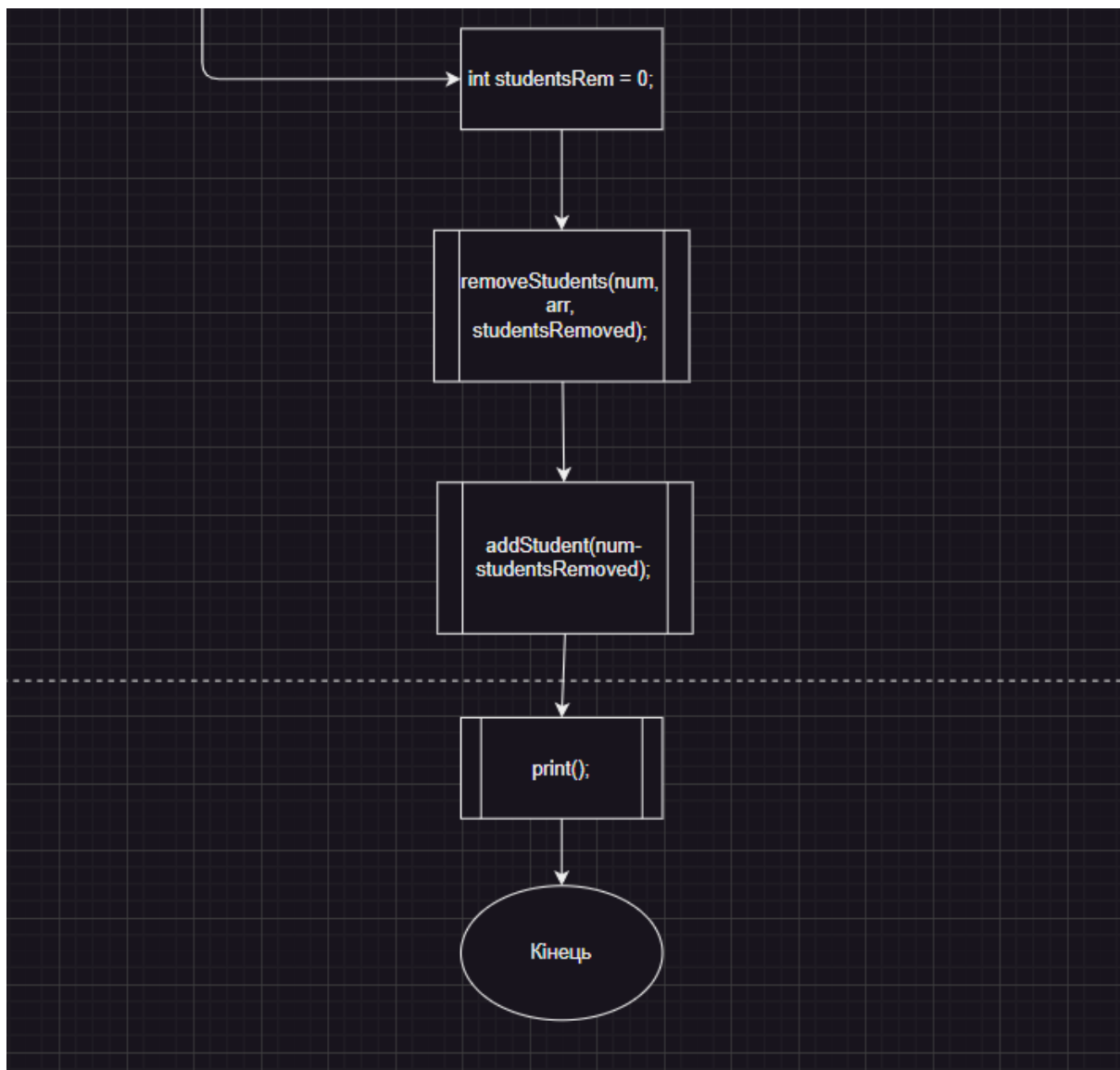
Окрім цього, дане завдання виявилось найскладнішим поміж усіх тому блок-схему я написав саме до нього. На написання блок-схеми я витратив годину.

Блок-схема до функції main:



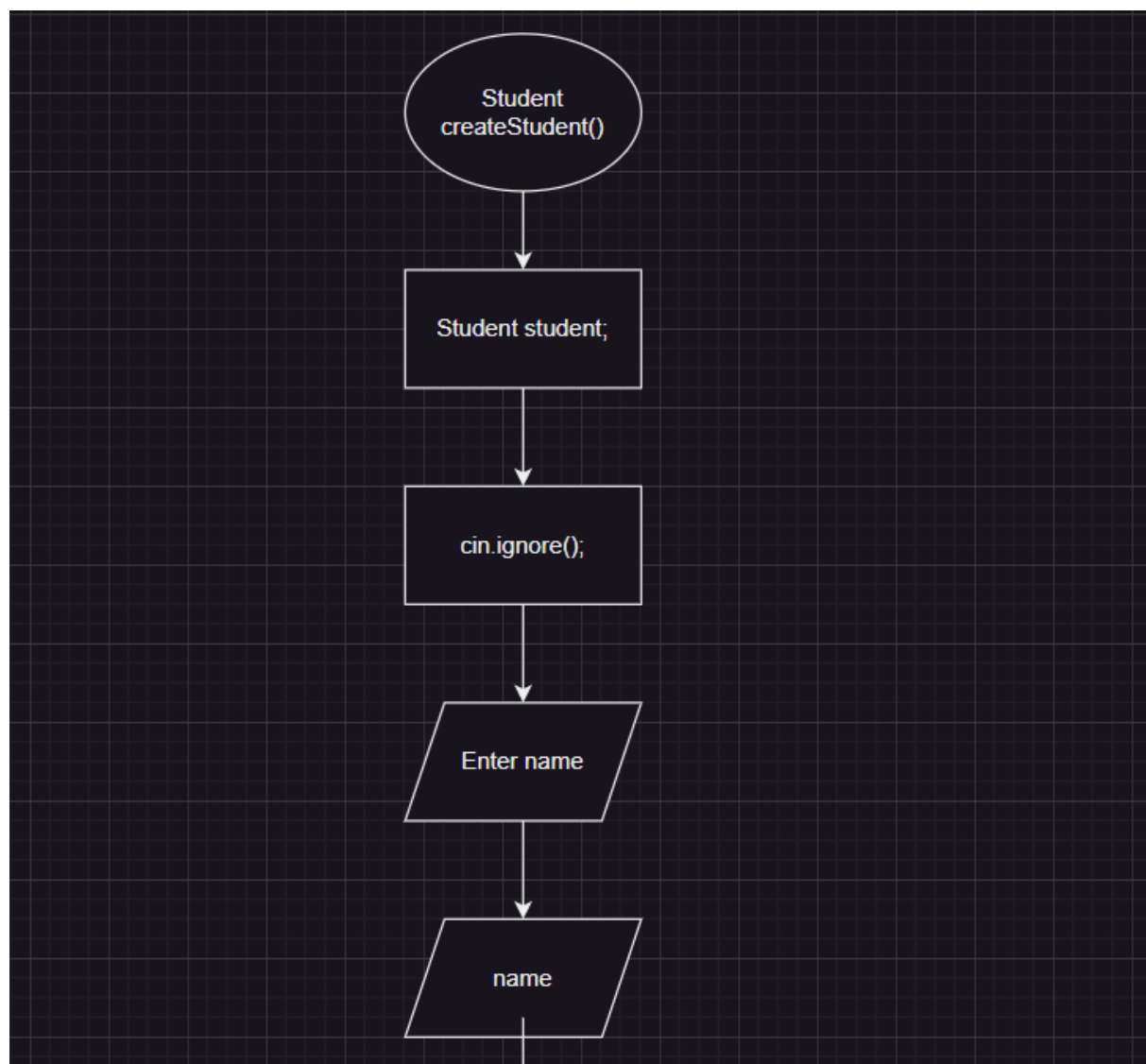


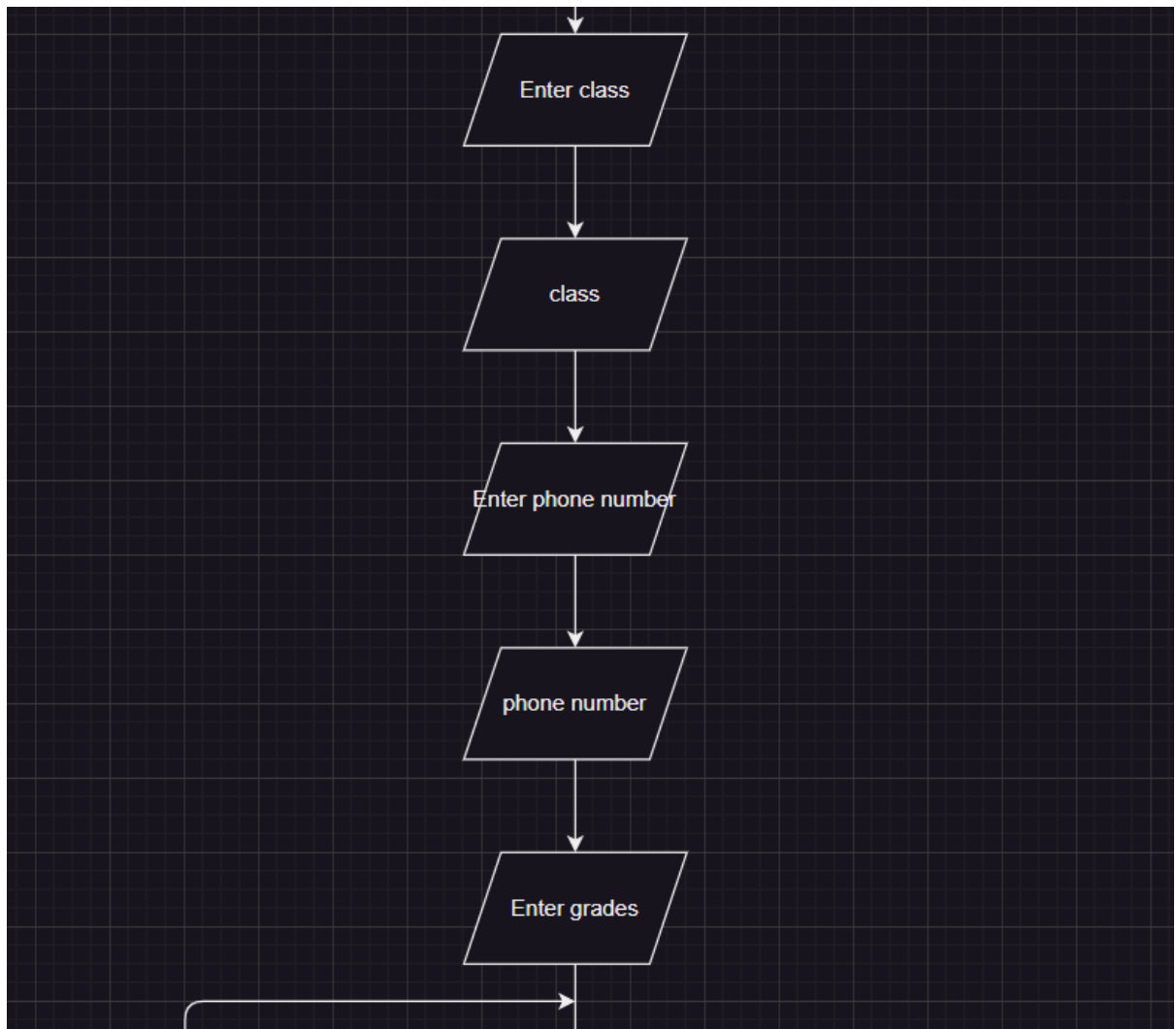


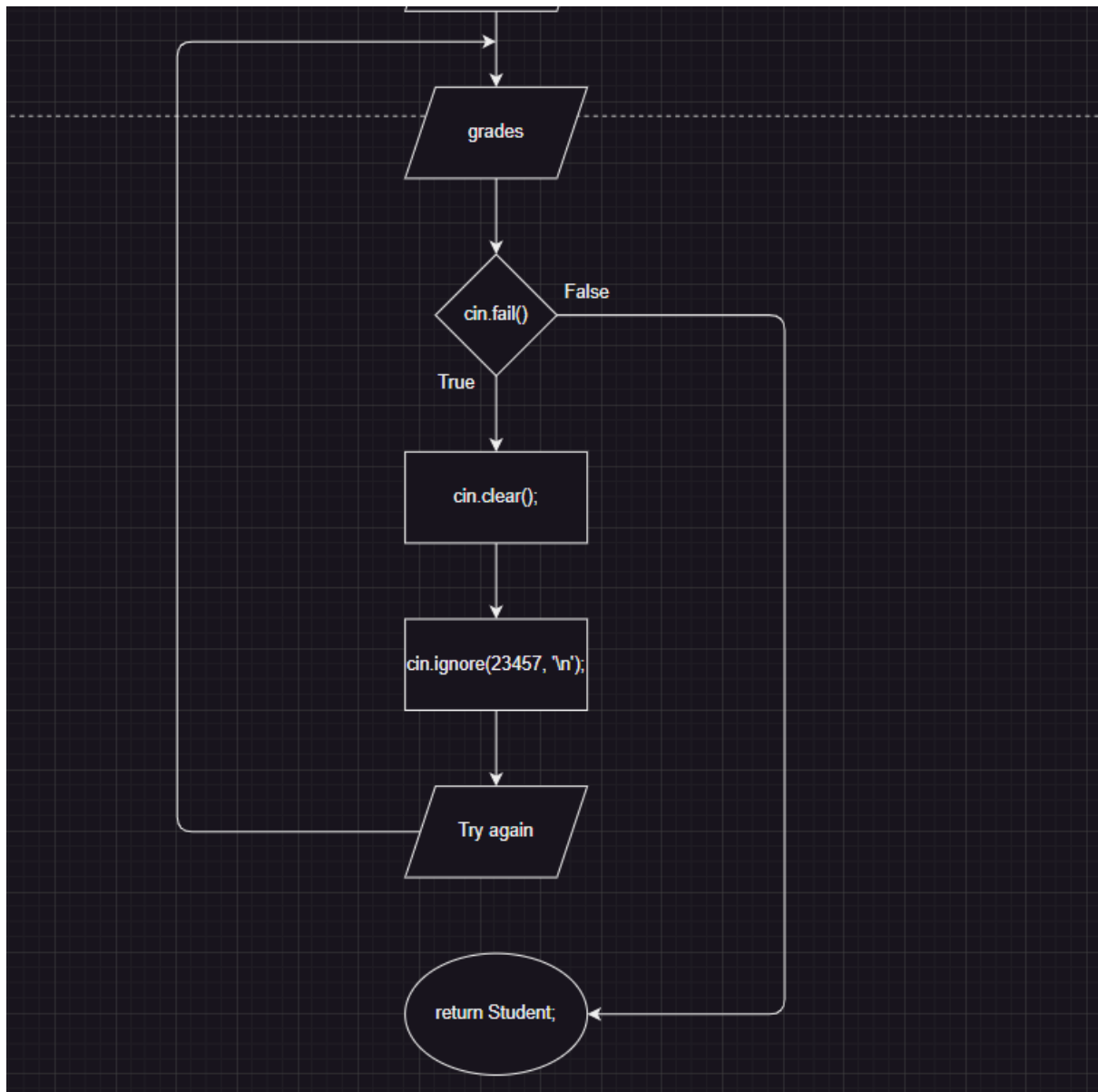


Блок-схема до функції `createStudent`:

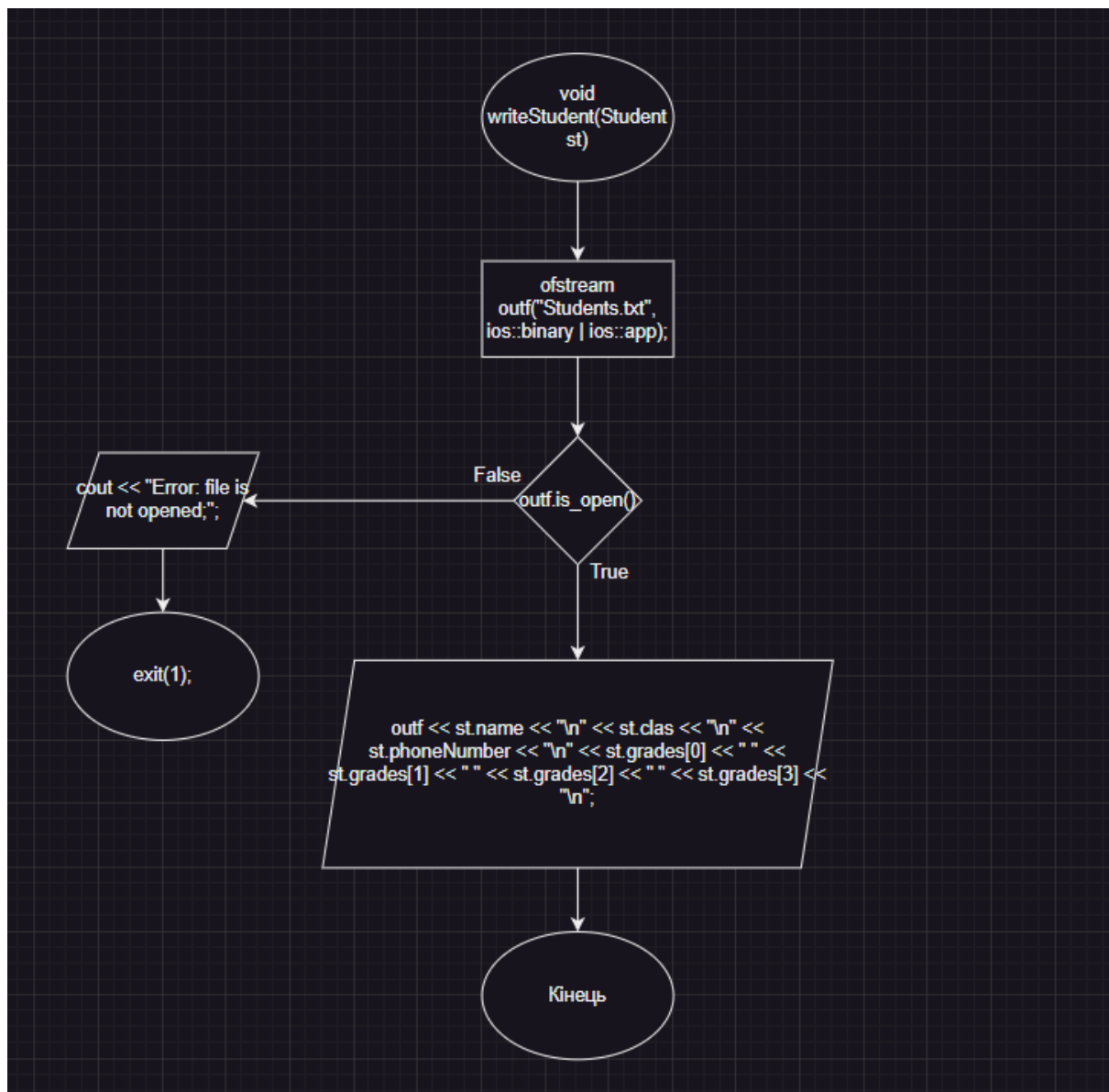




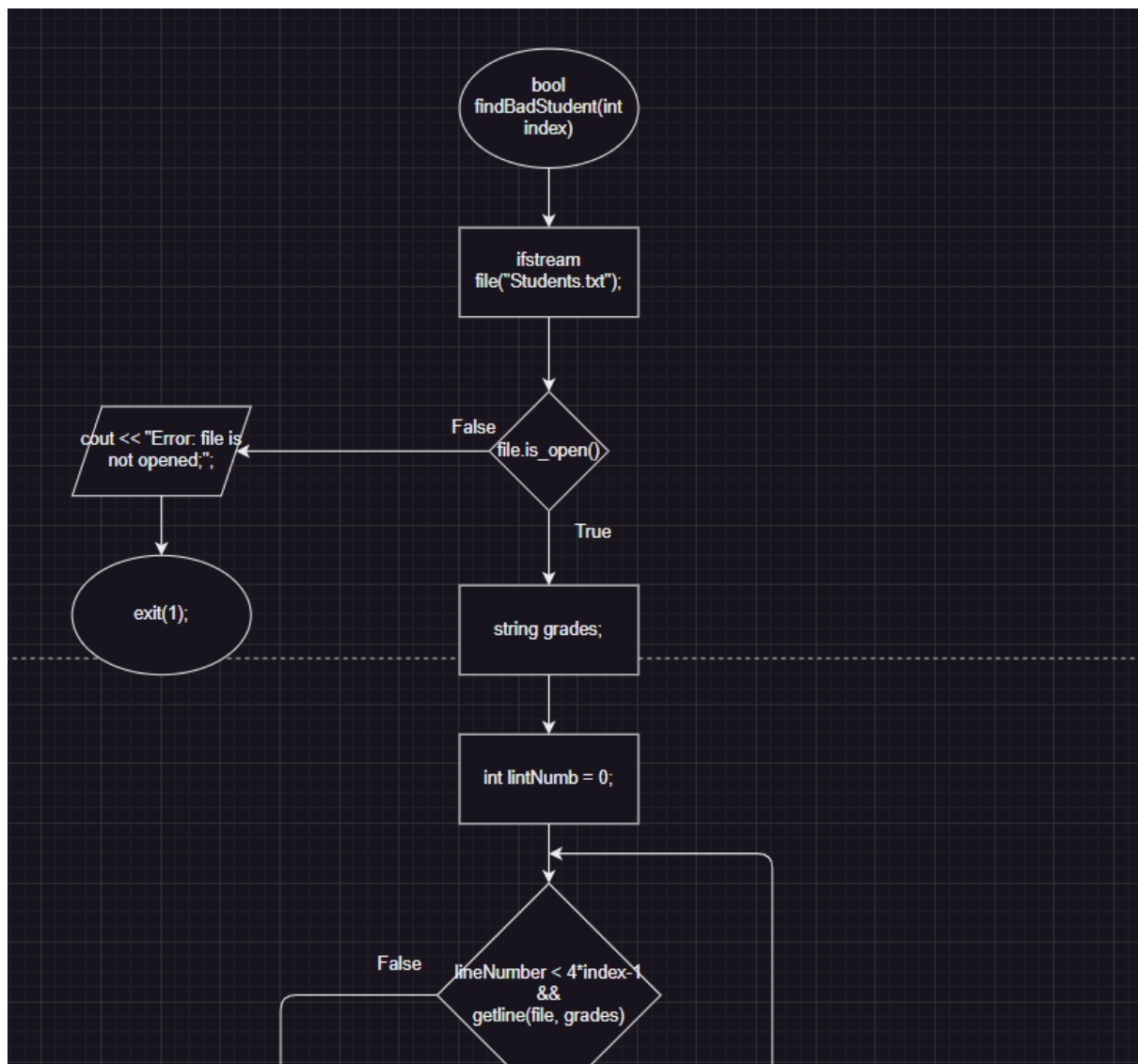


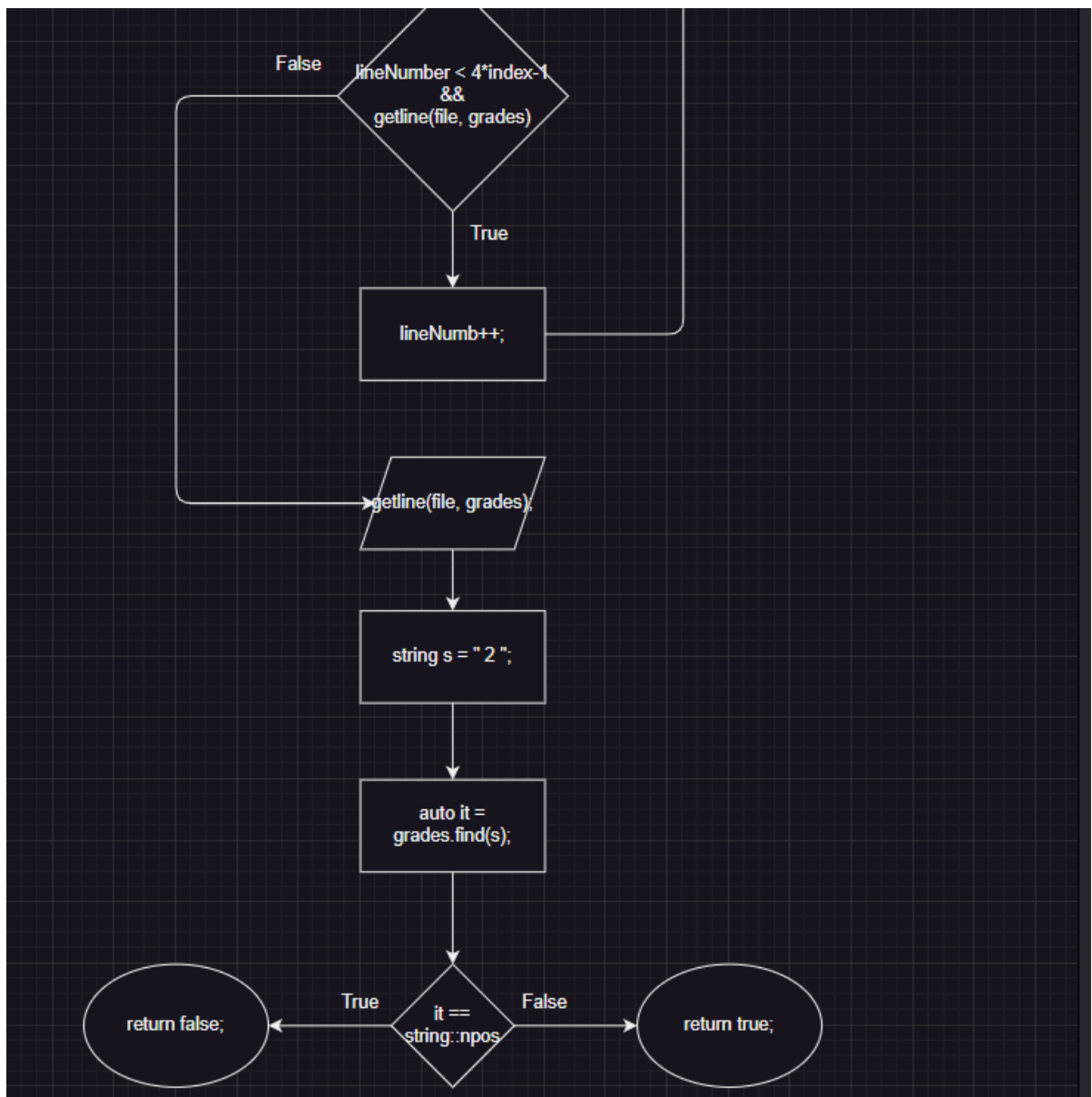


Блок-схема до функції `writeStudent`:

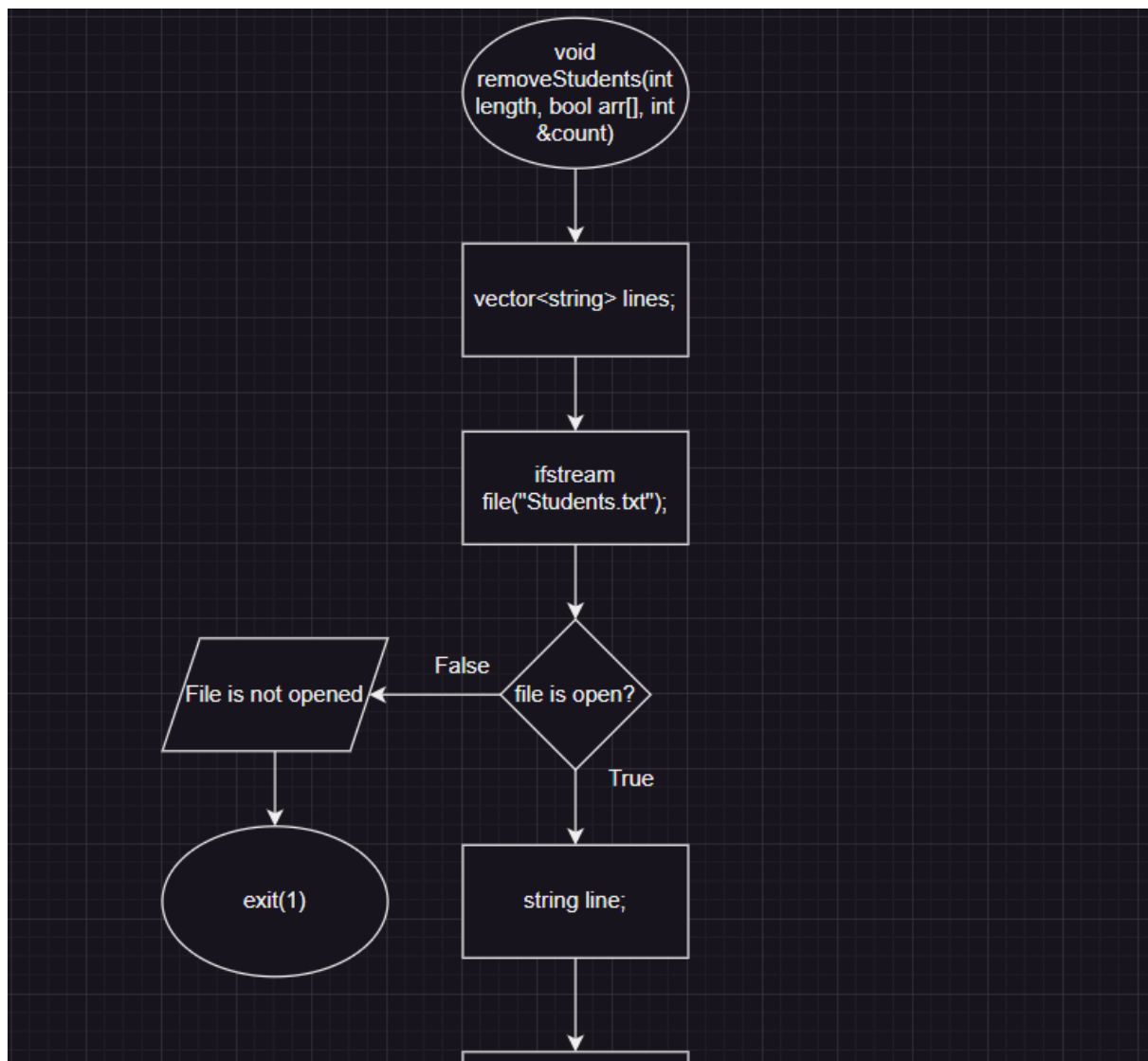


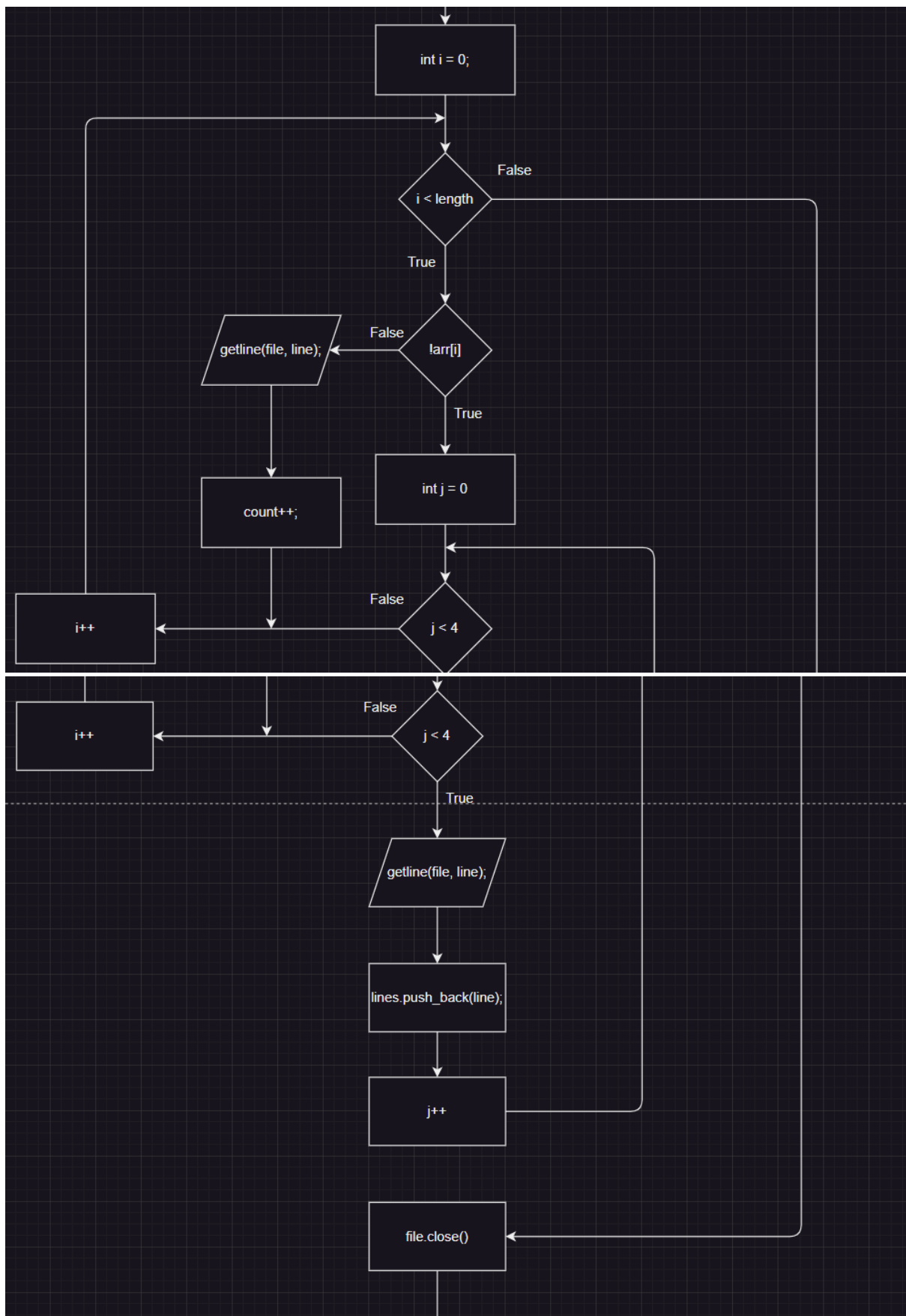
Блок-схема до функції `findBadStudent`:



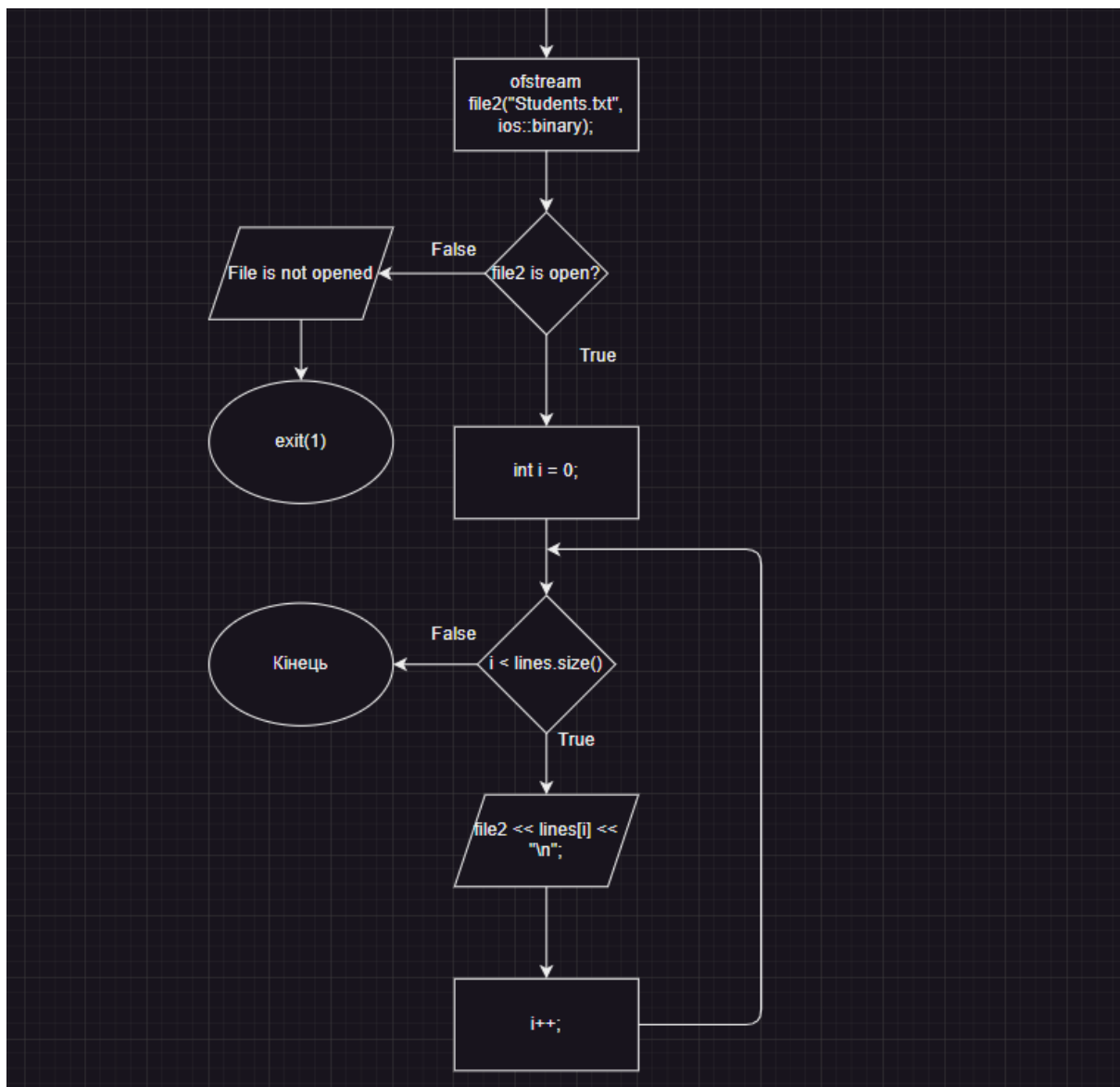


Блок - схема до функції removeStudents:

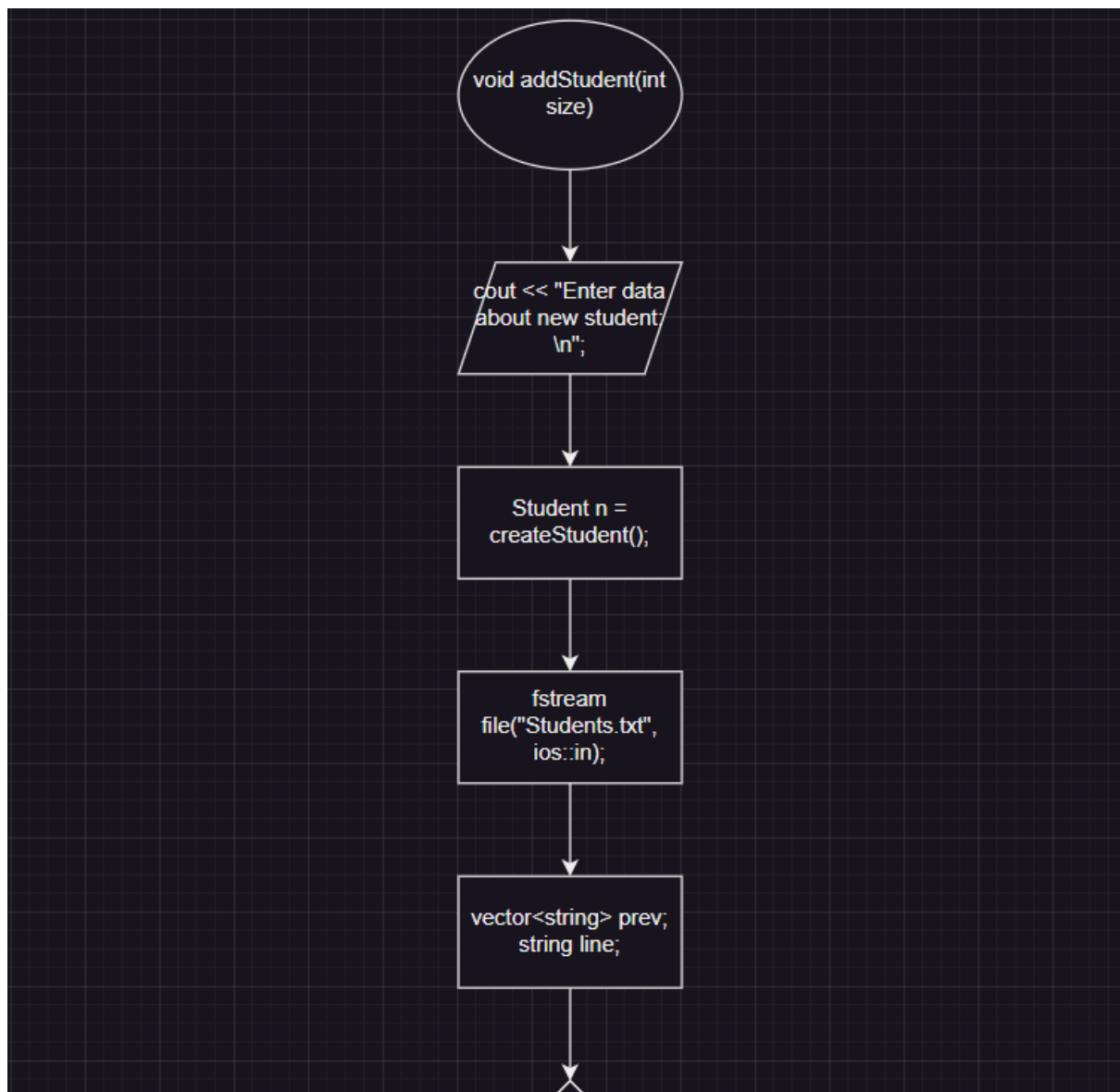


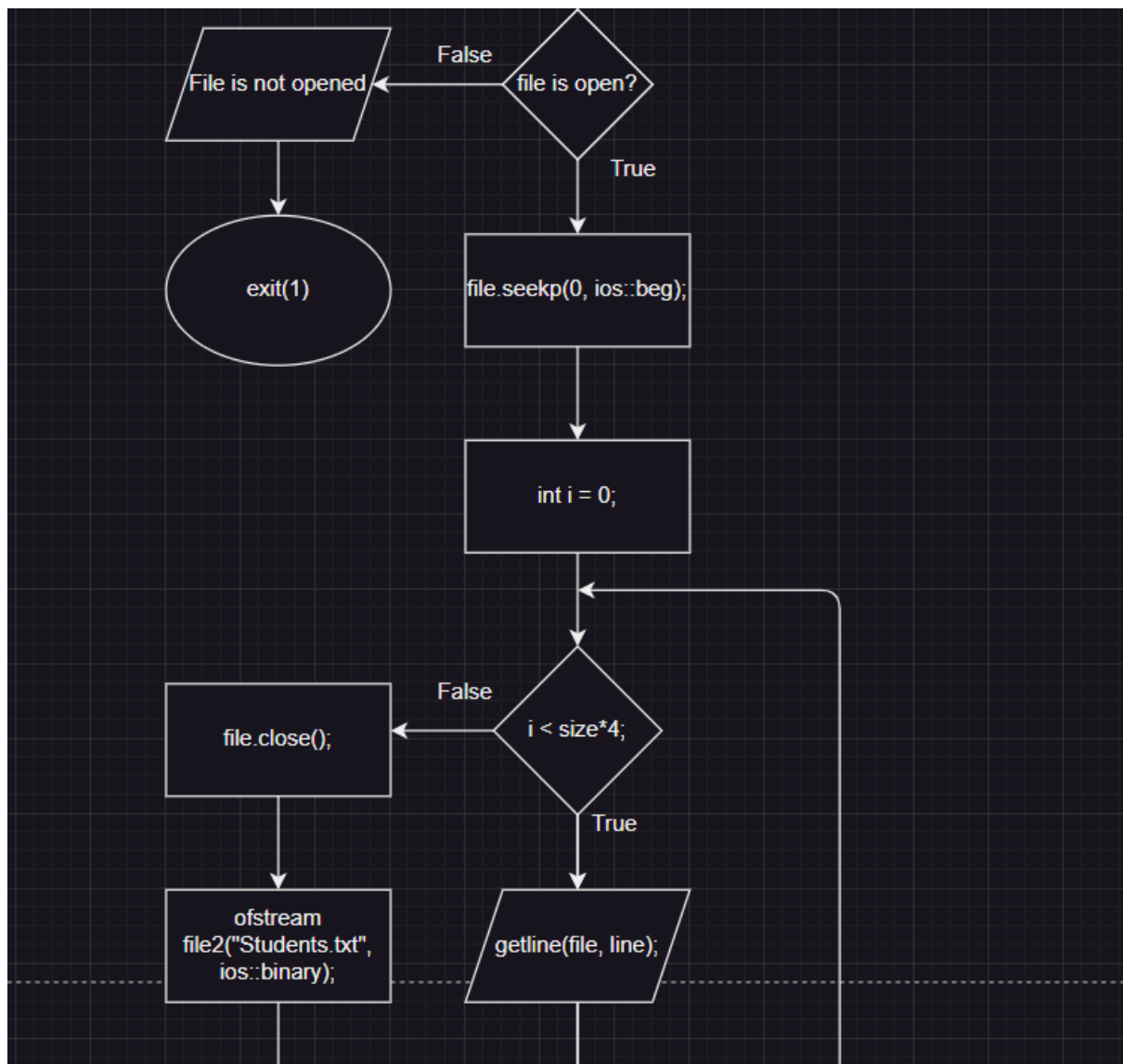


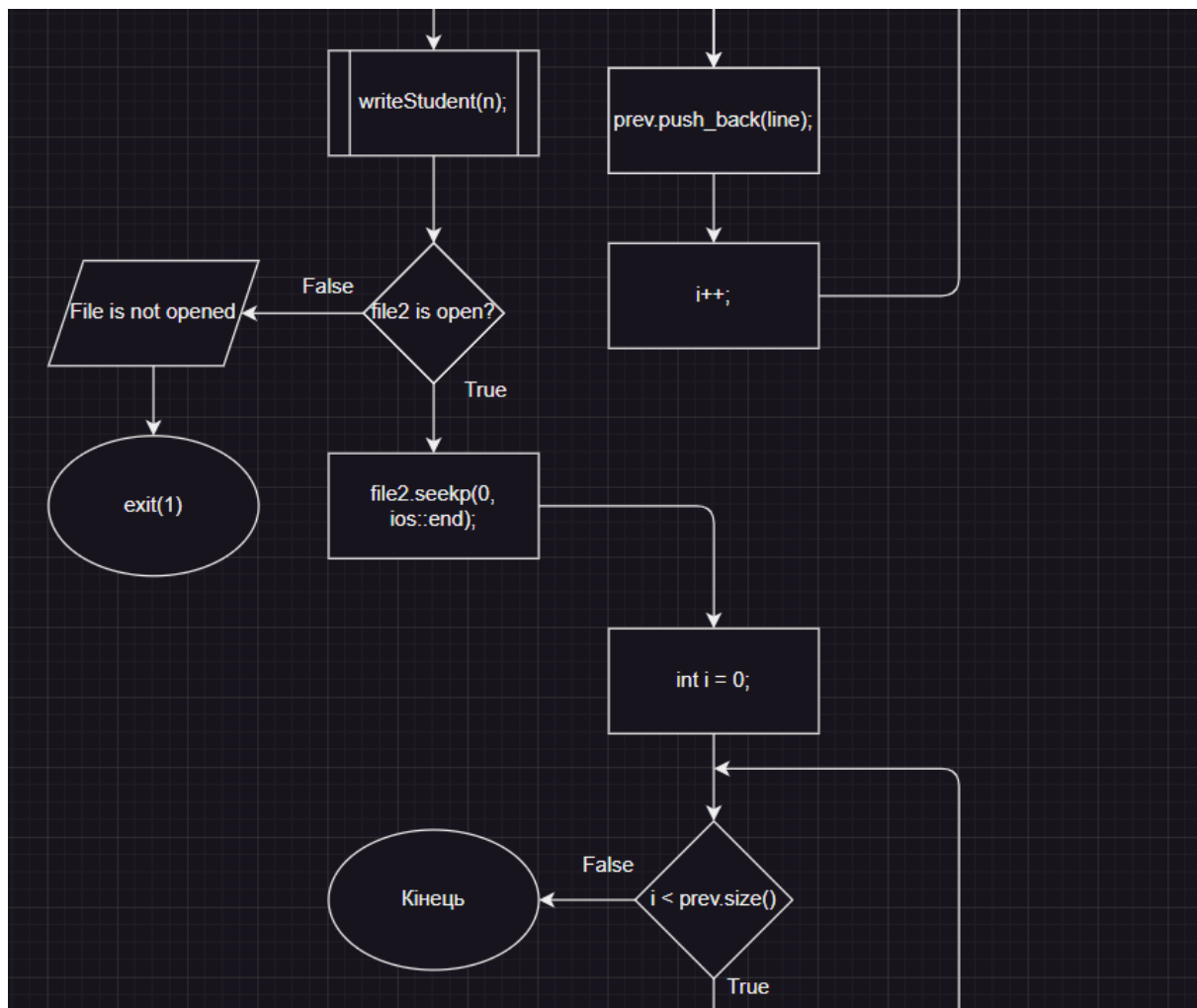


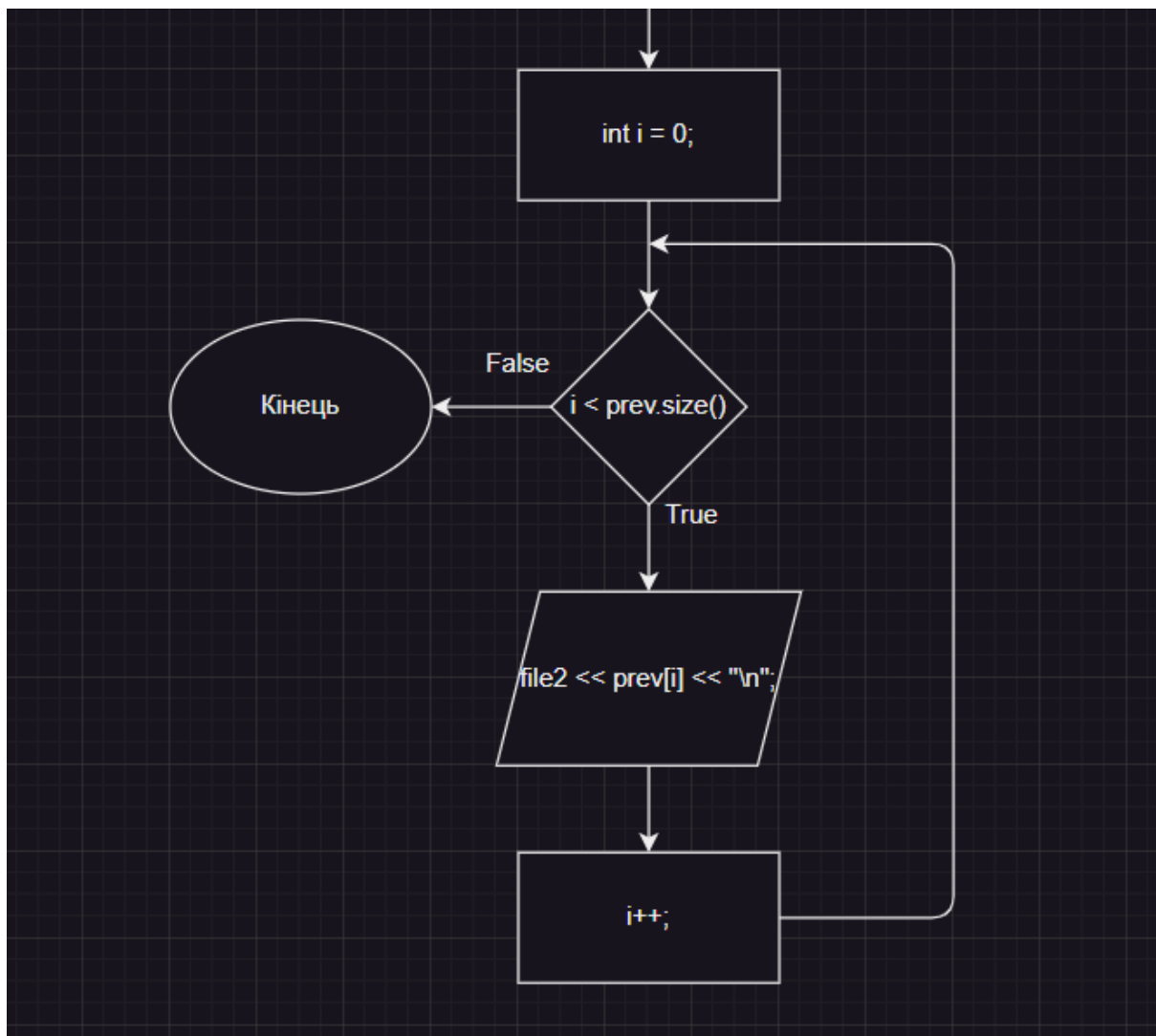


Блок-схема до функції addStudent:

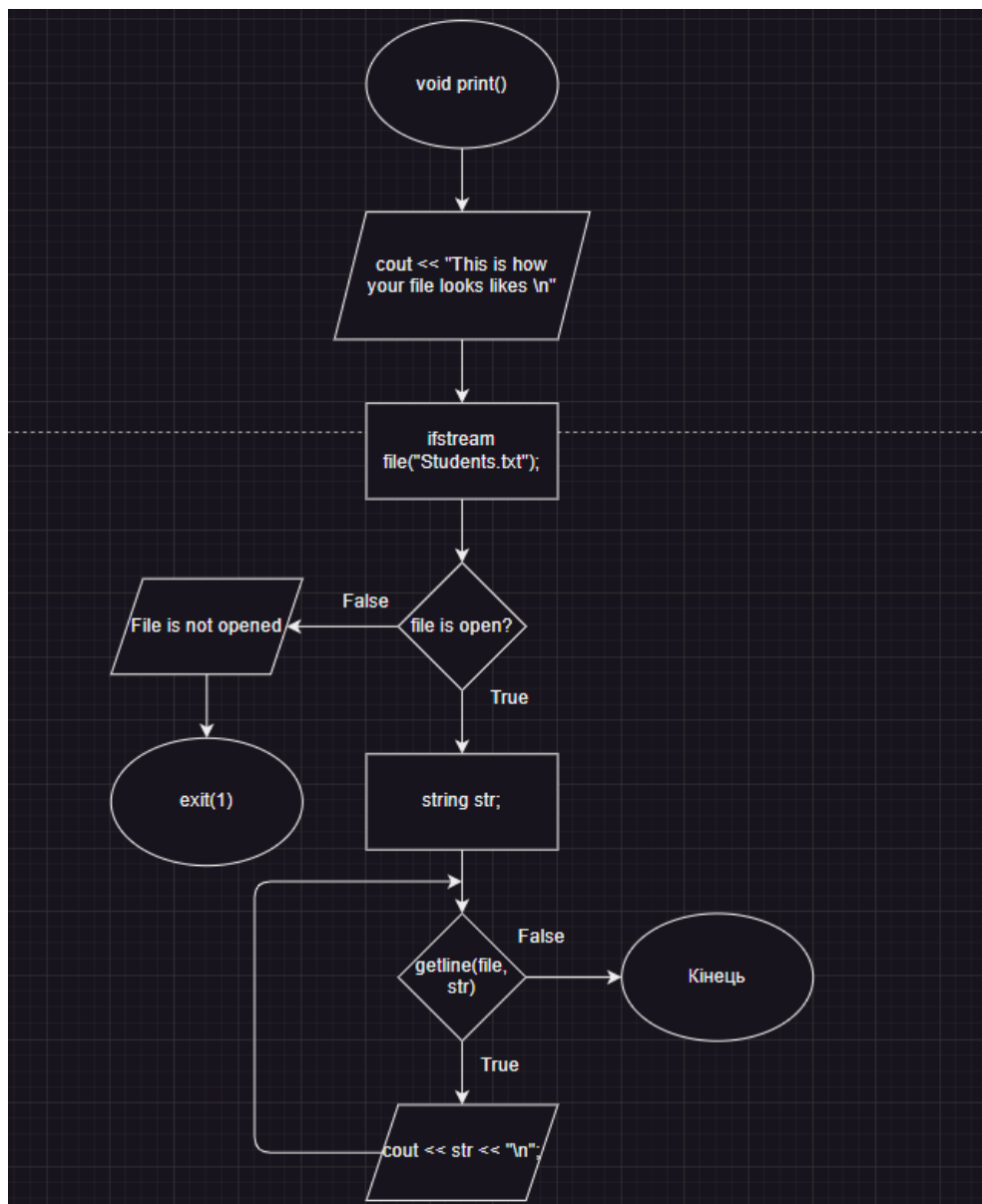








Блок-схема до функції print:



Програма:

```

1  #include <iostream>
2  #include <fstream>
3  #include <string>
4  #include <vector>
5  using namespace std;
6  /*
7   Знищити всі елементи, у яких є 2 хоча б з одного предмету, додати елемент у
8   початок файлу.
9   */
10
11  /* Приклади:
12  Ivanov Petro Oleksiyovych 10-B 098-123-4567 9 2 10 9
13  Petriv Andrii Vasylovych 11-C 067-890-1234 7 6 12 7
14  Sydorenko Mariia Ivanivna 9-A 095-456-7890 10 9 10 8
15  Koval Oksana Mykolaivna 10-A 050-321-6547 8 8 9 2
16  */
17
18  struct Student{
19      string name;
20      string clas;
21      string phoneNumber;
22      int grades[4];
23  };
24
25  Student createStudent() {
26      Student student;
27      cin.ignore();
28      cout << "Enter name: ";
29      getline(cin, student.name);
30      cout << "Enter class: ";
31      cin >> student.clas;
32      cout << "Enter phone number: ";
33      cin >> student.phoneNumber;
34      cout << "Enter grades: ";
35      cout << "Enter grades: ";
36      while(true) {
37          cin >> student.grades[0] >> student.grades[1] >> student.grades[2] >> student.grades[3];
38          if(cin.fail()) {
39              cin.clear();
40              cin.ignore(23457, '\n');
41              cout << "Try again: ";
42          } else break;
43      }
44      return student;
45  }
46
47  void writeStudent(Student st) {
48      ofstream outf("Students.txt", ios::binary | ios::app);
49      if(outf.is_open()) {
50          outf << st.name << "\n" << st.clas << "\n" << st.phoneNumber << "\n" <<
51              st.grades[0] << " " << st.grades[1] << " " << st.grades[2] << " " << st.grades[3] << "\n";
52      }
53      else {
54          cout << "Error: file is not opened;";
55          exit(1);
56      }
57  }
58
59  bool findBadStudent(int index) {
60      ifstream file("Students.txt");
61      if(file.is_open()) {
62          string grades;
63          int lineNumber = 0;
64          while (lineNumber < 4*index-1 && getline(file, grades)) {
65              lineNumber++;
66          }
67          getline(file, grades);
68          string s = " 2 ";
69          auto it = grades.find(s);
70          if(it != grades.end()) {
71              cout << "Bad student found at index " << index << endl;
72          }
73      }
74  }

```

```

69         if(it == string::npos) return false;
70         else return true;
71     } else {
72         cout << "Error: file is not opened;";
73         exit(1);
74     }
75 }
76
77 void removeStudents(int length, bool arr[], int &count) {
78     vector<string> lines;
79     ifstream file("Students.txt");
80     if(file.is_open()) {
81         string line;
82         for(int i = 0; i < length; i++) {
83             if(!arr[i]) {
84                 for(int j = 0; j < 4; j++) {
85                     getline(file, line);
86                     lines.push_back(line);
87                 }
88             }
89             else {
90                 getline(file, line);
91                 getline(file, line);
92                 getline(file, line);
93                 getline(file, line);
94                 count++;
95             }
96         }
97         file.close();
98     } else {
99         cout << "Error: file is not opened;";
100         exit(1);
101     }

```

```

106     }
107 } else {
108     cout << "Error: file is not opened;";
109     exit(1);
110 }
111 }
112
113 void addStudent(int size) {
114     cout << "Enter data about new student: \n";
115     Student n = createStudent();
116     fstream file("Students.txt", ios::in);
117     vector<string> prev;
118     string line;
119     if(file.is_open()) {
120         file.seekp(0, ios::beg);
121         for(int i = 0; i < size*4; i++) {
122             getline(file, line);
123             prev.push_back(line);
124         }
125         file.close();
126         ofstream file2("Students.txt", ios::binary);
127         writeStudent(n);
128         if(file2.is_open()) {
129             file2.seekp(0, ios::end);
130             for(int i = 0; i < prev.size(); i++) {
131                 file2 << prev[i] << "\n";
132             }

```



```

132         }
133     } else {
134         cout << "Error: file is not opened;";
135         exit(1);
136     }
137 } else {
138     cout << "Error: file is not opened;";
139     exit(1);
140 }
141 }
142
143 void print() {
144     cout << "This is how our file looks likes: \n";
145     ifstream file("Students.txt");
146     if(file.is_open()) {
147         string str;
148         while(getline(file, str)) {
149             cout << str << "\n";
150         }
151     } else {
152         cout << "Error: file is not opened;";
153         exit(1);
154     }
155 }
156

```

```

157 int main() {
158     cout << "How many students you want to create? ";
159     int num;
160     cin >> num;
161     ofstream file("Students.txt", ios::binary);
162     file.close();
163     for(int i = 0; i < num; i++) {
164         cout << "Enter data about " << i + 1 << " student:\n";
165         Student student = createStudent();
166         writeStudent(student);
167     }
168     bool arr[100];
169     for(int i = 0; i < num; i++) {
170         arr[i] = findBadStudent(i+1);
171     }
172     int studentsRemoved = 0;
173     removeStudents(num, arr, studentsRemoved);
174     addStudent(num-studentsRemoved);
175     print();
176     return 0;
177 }

```

Результат:

```
task_yevhen_kutel'makh } ; if ($?) { .\vns_lab_8_task_yevhen_kutel'makh }
How many students you want to create? 3
Enter data about 1 student:
Enter name: Ivanov Petro Oleksiyovych
Enter class: 10-B
Enter phone number: 098-123-4567
Enter grades: 9 2 10 9
Enter data about 2 student:
Enter name: Petriv Andrii Vasylovych
Enter class: 11-C
Enter phone number: 067-890-1234
Enter grades: 7 6 12 7
Enter data about 3 student:
Enter name: Sydorenko Mariia Ivanivna
Enter class: 9-A
Enter phone number: 095-456-7890
Enter grades: 10 9 10 8
Enter data about new student:
Enter name: Koval Oksana Mykolaivna
Enter class: 10-A
Enter phone number: 050-321-6547
Enter grades: 8 8 9 2
This is how our file looks likes:
Koval Oksana Mykolaivna
10-A
050-321-6547
8 8 9 2
Petriv Andrii Vasylovych
11-C
067-890-1234
7 6 12 7
Sydorenko Mariia Ivanivna
9-A
095-456-7890
10 9 10 8
PS C:\Users\kutel\.vscode\projects\uni\epic5>
```

Students.txt:

```
1 Koval Oksana Mykolaivna
2 10-A
3 050-321-6547
4 8 8 9 2
5 Petriv Andrii Vasylovych
6 11-C
7 067-890-1234
8 7 6 12 7
9 Sydorenko Mariia Ivanivna
10 9-A
11 095-456-7890
12 10 9 10 8
13
```

## Task 5 - Lab# programming: VNS Lab 9

Умова: Створити текстовий файл F1 не менше, ніж з 10 рядків і записати в нього інформацію. Виконати завдання.

Варіант 6: 1) Скопіювати з файлу F1 у файл F2 рядки, починаючи з N до K.

2) Підрахувати кількість приголосних букв у файлі F2.

Це завдання не таке тяжке як попереднє, тому я витратив на нього як і очікував 1.5 години

Програма:

```

1  #include <iostream>
2  #include <fstream>
3  #include <ctime>
4  #include <string>
5  #include <cstdlib>
6  using namespace std;
7
8  /*
9  1) Скопіювати з файлу F1 у файл F2 рядки, починаючи з N до K.
10 2) Підрахувати кількість приголосних букв у файлі F2.
11 */
12
13 int getRandomNumber(int min, int max)
14 {
15     static const double fraction = 1.0 / (static_cast<double>(RAND_MAX) + 1.0);
16     return static_cast<int>(rand() * fraction * (max - min + 1) + min);
17 }
18
19 string getRandomLine() {
20     string line = " ";
21     for(int i = 0; i < 10; i++) {
22         line[i] = getRandomNumber(97, 122);
23     }
24     return line;
25 }
26
27 int main() {
28     srand(static_cast<unsigned int>(time(0)));
29
30     fstream file("F1", ios::out | ios::trunc);
31     if(file.is_open()) {
32         for(int i = 0; i < 10; i++) {
33             string line = getRandomLine();
34             file << line << "\n";
35         }
36     } else {
37         cout << "Error: file F1 is not opened;";
38         exit(1);
39     }
40     file.close();
41
42     file.open("F1", ios::in);
43     if(!file.is_open()) {
44         cout << "Error: file F2 is not opened for reading;";
45         exit(1);
46     }
47     file.seekg(0, ios::beg);
48     int n = getRandomNumber(2, 4), k = getRandomNumber(5, 9);
49     cout << "We will copy lines from " << n << " to " << k;
50
51     fstream file2("F2", ios::out | ios::trunc);
52     if(file2.is_open()) {
53         for(int i = 1; i <= k; i++) {
54             string copy;
55             getline(file, copy);
56             if(i >= n)
57                 file2 << copy << "\n";
58         }
59     } else {
60         cout << "Error: file F2 is not opened;";
61         exit(1);
62     }
63     file.close();
64     file2.close();
65

```

```

66     file2.open("F2", ios::in);
67     if(!file2.is_open()) {
68         cout << "Error: file F2 is not opened for reading;";
69         exit(1);
70     }
71
72     char c;
73     int num = 0;
74     while(file2.get(c)) {
75         switch(c) {
76             case 'a':
77             case 'o':
78             case 'i':
79             case 'u':
80             case 'e':
81             case '\n': break;
82             default: num++; break;
83         }
84     }
85
86     cout << "\nWe have " << num << " consonant letters in F2";
87
88     file2.close();
89     return 0;
90 }

```

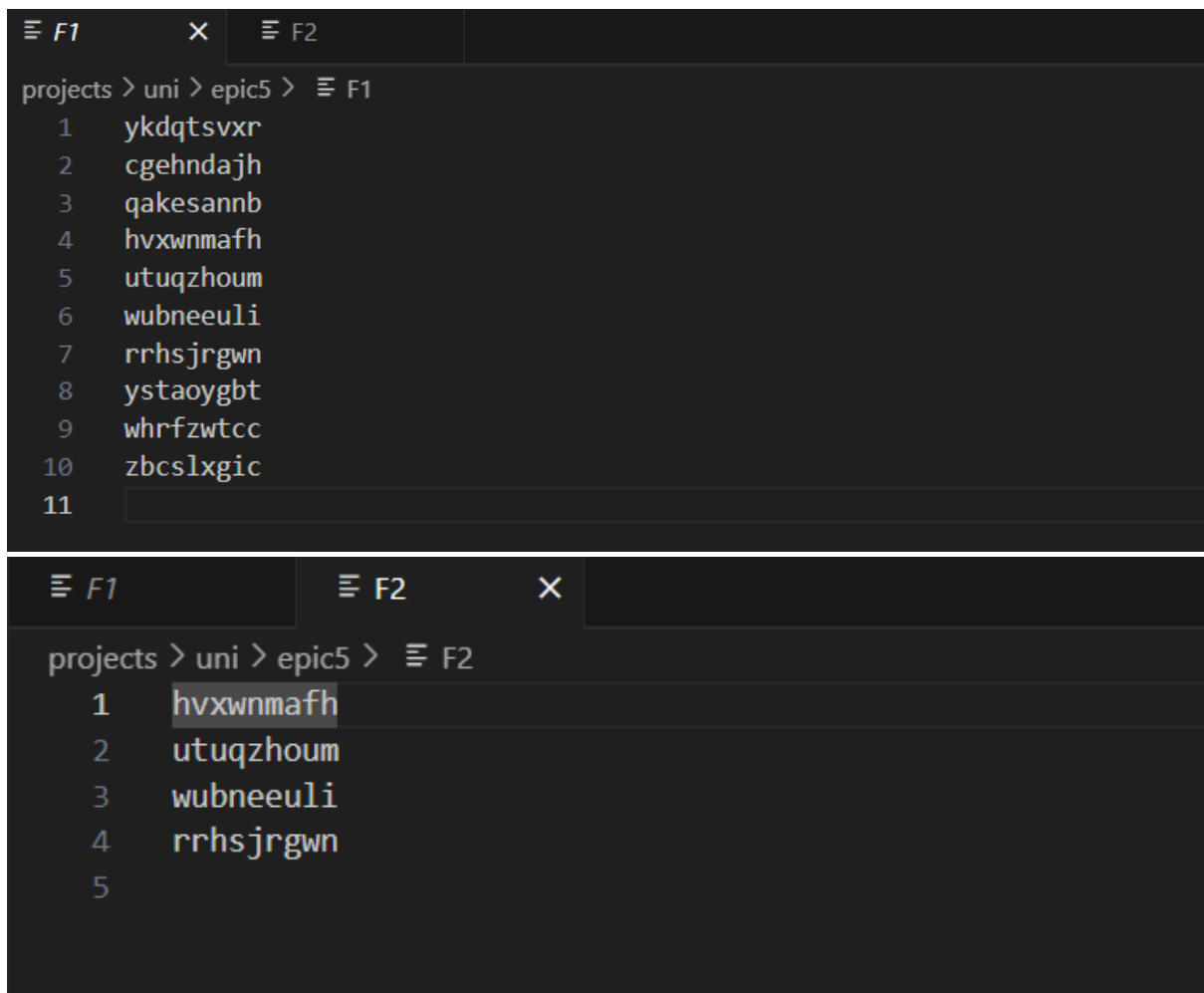
Результат:

```

We will copy lines from 4 to 7
We have 26 consonant letters in F2
PS C:\Users\kutel\.vscode\projects\uni\epic5>

```

F1 i F2:



## Task 6 - Lab# programming: Algotester Lab 4

### 1) Розв'язок з використанням STL:

#### Умова:

Вам дано масив  $a$  з  $N$  цілих чисел.

Спочатку видаліть масиву  $a$  усі елементи що повторюються, наприклад масив  $[1, 3, 3, 4]$  має перетворитися у  $[1, 3, 4]$ .

Після цього оберніть посортовану версію масиву  $a$  на  $K$ , тобто при  $K = 3$  масив  $[1, 2, 3, 4, 5, 6, 7]$  перетвориться на  $[4, 5, 6, 7, 1, 2, 3]$ .

Виведіть результат.

#### Input

У першому рядку цілі числа  $N$  та  $K$

У другому рядку  $N$  цілих чисел - елементи масиву  $a$

#### Output

У першому рядку ціле число  $N$  - розмір множини  $a$

У наступному рядку  $N$  цілих чисел - множина  $a$

| Created    | Compiler | Result         | Time (sec.) | Memory (MiB) | Actions              |
|------------|----------|----------------|-------------|--------------|----------------------|
| 6 days ago | C++ 23   | Accepted       | 0.003       | 1.277        | <a href="#">View</a> |
| 6 days ago | C++ 23   | Wrong Answer 1 | 0.003       | 1.078        | <a href="#">View</a> |
| 6 days ago | C++ 23   | Wrong Answer 1 | 0.003       | 0.918        | <a href="#">View</a> |

Завдання не було складним, адже можна використовувати STL, я витратив на нього 20 хв, та дізнався нову інформацію про контейнери та алгоритми стандартної бібліотеки.

Програма:

```

1 #include <iostream>
2 #include <algorithm>
3 #include <vector>
4 #include <set>
5 using namespace std;
6
7 //v2 STL
8
9 int main() {
10     int N, K;
11     cin >> N >> K;
12     int num;
13     set<int> arr;
14     for(int i = 0; i < N; i++) {
15         cin >> num;
16         arr.insert(num);
17     }
18     vector<int> ar;
19     for(int el : arr)
20         ar.push_back(el);
21     while(K > ar.size()) {
22         K -= ar.size();
23     }
24     vector<int>::iterator it = ar.begin() + K;
25     rotate(ar.begin(), it, ar.end());
26     cout << ar.size() << "\n";
27     for(int el : ar)
28         cout << el << " ";
29     return 0;
30 }

```

Результат:

```

5 7
1 9 2 3 1
4
9 1 2 3
PS C:\Users\kutel\.vscode\projects\uni\epic>

```

## 2) Розв'язок без використання STL:

Умова:

### Lab 4v3

Limits: 2 sec., 256 MiB

Вам дано масив, який складається з  $N$  додатніх цілих чисел.

Ваше завдання - розділити його на три частини, по остачі від ділення на 3, по зростанню остачі (тобто спочатку йдуть числа, у яких остача 0, далі числа з остачею 1 і тоді нарешті числа з остачею 2).

Далі необхідно ті елементи, остача від ділення на 3 яких парна посортувати по зростанню, а ті, у яких остача 1 - по спаданню.

Після цього видаліть усі дублікати з масиву.

Виведіть результуючий масив.

### Input

У першому рядку  $N$  - кількість чисел.

У другому рядку  $N$  чисел  $a_i$  - елементи масиву.

### Output

У першому рядку  $M$  - кількість чисел у масиву

У другому рядку  $M$  посортованих за умовою чисел.

| Created    | Compiler | Result   | Time (sec.) | Memory (MiB) | Actions              |
|------------|----------|----------|-------------|--------------|----------------------|
| 6 days ago | C++ 23   | Accepted | 0.003       | 1.242        | <a href="#">View</a> |

Це завдання вже складніше, саме по собі (порівнюючи з попереднім) та тому що тут не можна використовувати STL. Очікував зробити його за 30 хв, проте витратив годину.

Програма:

```

1  #include <iostream>
2  using namespace std;
3
4  //v3 NSTL
5
6  int checkD(int num) {
7      return num % 3;
8  }
9
10 void createMas(int s, int start[], int ar0[], int ar1[], int ar2[]) {
11     int s1 = 0, s2 = 0, s0 = 0;
12     for(int i = 0; i < s; i++) {
13         switch (checkD(start[i])) {
14             case 0: ar0[s0] = start[i]; s0++; break;
15             case 1: ar1[s1] = start[i]; s1++; break;
16             case 2: ar2[s2] = start[i]; s2++; break;
17         }
18     }
19 }
20
21 void Quicksort(int *arr, int first, int last) {
22     if (first >= last) return;
23     int med = arr[(first + last) / 2];
24     int i = first, j = last;
25
26     while (i <= j) {
27         while (arr[i] < med) i++;
28         while (arr[j] > med) j--;
29         if (i <= j) {
30             swap(arr[i], arr[j]);
31             i++; j--;
32         }
33     }
34
35     if (first < j) Quicksort(arr, first, j);
36     if (i < last) Quicksort(arr, i, last);
37 }

```

```

39 void Quicksort(int *arr, int first, int last, char) {
40     if (first >= last) return;
41     int med = arr[(first + last) / 2];
42     int i = first, j = last;
43
44     while (i <= j) {
45         while (arr[i] > med) i++;
46         while (arr[j] < med) j--;
47         if (i <= j) {
48             swap(arr[i], arr[j]);
49             i++; j--;
50         }
51     }
52
53     if (first < j) Quicksort(arr, first, j, 'c');
54     if (i < last) Quicksort(arr, i, last, 'c');
55 }
56
57 bool checkifpresented(int *arr, int size, int key) {
58     for(int i = 0; i < size; i++) {
59         if(arr[i]==key) return true;
60     }
61     return false;
62 }
63
64 int delEl(int *arr, int *arrres, int size) {
65     int truesize = 1;
66     arrres[0] = arr[0];
67     for(int i = 1; i < size; i++) {
68         if(!checkifpresented(arrres, truesize, arr[i])) {
69             arrres[truesize] = arr[i];
70             truesize++;
71         }
72     }
73     return truesize;
74 }

```

```

76  int main() {
77      int N;
78      cin >> N;
79      int *arr = new int[N];
80      int os1 = 0, os2 = 0, os0 = 0;
81      for(int i = 0; i < N; i++) {
82          cin >> arr[i];
83          switch (checkD(arr[i])) {
84              case 0: os0++; break;
85              case 1: os1++; break;
86              case 2: os2++; break;
87          }
88      }
89      int *arr0 = new int[os0];
90      int *arr1 = new int[os1];
91      int *arr2 = new int[os2];
92      createMas(N, arr, arr0, arr1, arr2);
93      Quicksort(arr2, 0, os2-1);
94      Quicksort(arr1, 0, os1-1, 'c');
95      Quicksort(arr0, 0, os0-1);
96      int counter = 0;
97      for(; counter < os0; counter++) {
98          arr[counter] = arr0[counter];
99      }
100     for(; counter < os0 + os1; counter++) {
101         arr[counter] = arr1[counter-os0];
102     }
103     for(; counter < N; counter++) {
104         arr[counter] = arr2[counter-os0-os1];
105     }

106     int *arrres = new int[N];
107     int s = delEl(arr, arrres, N);
108     cout << s << "\n";
109     for(int i = 0; i < s; i++)
110         cout << arrres[i] << " ";
111     delete[] arr0;
112     delete[] arr1;
113     delete[] arr2;
114     delete[] arr;
115     delete[] arrres;
116     arr = nullptr;
117     arrres = nullptr;
118     arr0 = nullptr;
119     arr1 = nullptr;
120     arr2 = nullptr;
121     return 0;
122 }

```

Результат:



```
7
1 3 23 8 9 7 7
6
3 9 7 1 8 23
PS C:\Users\kute1\.vscode\projects\uni\epic5>
```

## Task 7 - Lab# programming: Algotester Lab 6

Умова:

### Lab 6v1

*Limits: 2 sec., 256 MB*

Вам дано  $N$  слів та число  $K$ .

Ваше завдання перерахувати букви в словах, які зустрічаються в тексті більше-рівне ніж  $K$  разів (саме слово, не буква!).

Великі та маленькі букви вважаються однаковими, виводити необхідно малі, посортовані від останньої до першої у алфавіті. Букву потрібно виводити лише один раз.

У випадку якщо таких букв немає - вивести "Empty!".

### Input

Цілі числа  $N$  та  $K$  - загальна кількість слів та мінімальна кількість слів щоб враховувати букви цього слова в результаті.

$N$  стрічок  $s$

### Output

У першому рядку ціле число  $M$  - кількість унікальних букв

У другому рядку унікальні букви через пробіли

Завдання не було надто складним, тому я впорався за 30 хв.

Програма:

```

1  #include <iostream>
2  #include <string>
3  #include <algorithm>
4  #include <vector>
5  #include <map>
6  using namespace std;
7
8  void lowerCase(string &str) {
9      int len = str.size();
10     for(int letter = 0; letter < len; letter++) {
11         if(str[letter] >= 65 && str[letter] <= 90)
12             str[letter] += 32;
13     }
14 }
15
16 int main() {
17     int N, K;
18     cin >> N >> K;
19     cin.ignore(32000, '\n');
20     vector<string> str(N);
21     for(int i = 0; i < N; i++) {
22         getline(cin, str[i]);
23         lowerCase(str[i]);
24     }
25
26     map<string, int> words;
27     for (const string &s : str) {
28         words[s]++;
29     }
30
31     vector<bool> isPresent(26, false);
32     for (const auto &pair : words) {
33         if (pair.second >= K) {
34             for (char ch : pair.first) {
35                 isPresent[ch - 'a'] = true;
36             }
37         }
38     }
39
40     vector<char> let;
41     for (int i = 25; i >= 0; i--) {
42         if (isPresent[i]) {
43             let.push_back('a' + i);
44         }
45     }
46
47     if(let.size()==0)
48         cout << "Empty!";
49     else {
50         cout << let.size() << "\n";
51         for(char el : let)
52             cout << el << " ";
53     }
54     return 0;
55 }

```

Результат:

```

0 algotester_lab_6_variant_1_yevhen_kutechakn } ; if ($?) { .\algotester_lab_6_variant_1_yevhen_kutechakn }
5 2
stugna
neptune
grim
oplol
Grim
4
r m i g
PS C:\Users\kutel\.vscode\projects\uni\epic5>

```

## Task 8 - Practice# programming: Class Practice Task

Завдання: Реалізувати функцію створення файлу і запису в нього даних:

```
enum FileOpResult { Success, Failure, ... };
```

```
FileOpResult write_to_file(char *name, char *content);
```

Умови задачі:

- створити файл із заданим ім'ям; якщо файл існує – перезаписати його вміст
- написати код стійкий до різних варіантів вхідних параметрів

- name – ім'я, може не включати шлях
- записати у файл вміст стрічки content, прочитати content із стандартного вводу
- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, запису даних, чи закриття файла.

*Реалізувати функцію створення файла і запису в нього даних:*

```
enum FileOpResult { Success, Failure, ... };
```

```
FileOpResult copy_file(char *file_from, char *file_to);
```

*Умови задачі:*

- копіювати вміст файла з ім'ям file\_from у файл з ім'ям file\_to; написати код стійкий до різних варіантів вхідних параметрів, обробити всі можливі варіанти відсутності одного з файлів
- file\_from, file\_to – можуть бути повним або відносним шляхом
- повернути статус операції: Success – все пройшло успішно, Failure – файл не вдалося створити, або збій операції відкриття, читання чи запису даних, закриття файла.

Я очікував зробити це завдання за 50 хв, вийшло за 70.

Програма:

```

1  #include <iostream>
2  #include <fstream>
3  #include <cstring>
4  #include <string>
5  using namespace std;
6
7  enum FileOpResult {
8      Failure,
9      Success,
10 };
11
12 FileOpResult write_to_file(char *name, char *content, bool n = false) {
13     ofstream file;
14     if (n) file.open(name, ios::app);
15     else file.open(name);
16     if(file.is_open()) {
17         file << content << "\n";
18         file.close();
19         ifstream file(name);
20         if(file.is_open()) {
21             char s[100];
22             bool found = false;
23             while(file.getline(s, sizeof(s))) {
24                 if(strcmp(s, content)==0) {
25                     found = true;
26                     break;
27                 }
28             }
29             if(found) return Success;
30             cerr << "Error: file's text doesn't match ours\n";
31             return Failure;
32         } else {
33             cerr << "Error: file couldn't get opened for reading";
34             return Failure;
35         }
36     }
37 }
38
39 }
40
41
42 FileOpResult copy_file(char *file_from, char *file_to) {
43     ifstream file1(file_from);
44     ofstream f(file_to);
45     f.close();
46     fstream file2(file_to);
47     if(file1.is_open() && file2.is_open()) {
48         string str;
49         while(getline(file1, str))
50             file2 << str << "\n";
51         file2.close(); file1.close();
52         file2.open(file_to, ios::in);
53         file2.open(file_to, ios::in);
54         if(file2.is_open()) {
55             file2.seekg(0, ios::beg);
56             file1.seekg(0, ios::beg);
57             string s1, s2;
58             while(getline(file1, s1) || getline(file2, s2)) {
59                 if(s1!=s2) {
60                     cerr << "Data of files isn't the same";
61                     return Failure;
62                 }
63                 s1 = " ";
64                 s2 = " ";
65             }
66             return Success;
67         }
68     }
69 }

```

```

36     } else {
37         cerr << "Error: file couldn't get opened for writing";
38         return Failure;
39     }
40 }
41
42 FileOpResult copy_file(char *file_from, char *file_to) {
43     ifstream file1(file_from);
44     ofstream f(file_to);
45     f.close();
46     fstream file2(file_to);
47     if(file1.is_open() && file2.is_open()) {
48         string str;
49         while(getline(file1, str))
50             file2 << str << "\n";
51         file2.close(); file1.close();
52         file2.open(file_to, ios::in);
53         file2.open(file_to, ios::in);
54         if(file2.is_open()) {
55             file2.seekg(0, ios::beg);
56             file1.seekg(0, ios::beg);
57             string s1, s2;
58             while(getline(file1, s1) || getline(file2, s2)) {
59                 if(s1!=s2) {
60                     cerr << "Data of files isn't the same";
61                     return Failure;
62                 }
63                 s1 = " ";
64                 s2 = " ";
65             }
66             return Success;
67         }
68     }
69 }

```

```

67     }
68     else {
69         cerr << "couldn't open file2 again";
70         return Failure;
71     }
72 } else {
73     cerr << "Couldn't open one of files";
74     return Failure;
75 }
76 }
77
78 int main() {
79     cout << "Hello, what do you want to do today?\nBut firstly you have to create a file\n";
80     char filename[100];
81     cout << "Enter a filename: ";
82     cin.getline(filename, 99);
83     char str[100];
84     cout << "Enter text you want to write into your file: ";
85     cin.getline(str, 99);
86     if(write_to_file(filename, str)) {
87         cout << "Writing file went smoosely\n";
88     } else exit(1);
89     cout << "Now you can either add new string into your file or new file, or copy-paste from one file to another\n";
90     char c;
91     while(true) {
92         cout << "Enter m for creating new file, a for adding new strings, c for copying, e for exit: ";
93         cin >> c;
94         cin.ignore();
95         switch(c) {
96             case 'e':
97                 return 0;

```

```

98             case 'm':{
99                 char filename[100];
100                 cout << "Enter a filename: ";
101                 cin.getline(filename, 99);
102                 char str[100];
103                 cout << "Enter text you want to write into your file: ";
104                 cin.getline(str, 99);
105                 if(write_to_file(filename, str)) {
106                     cout << "Writing file went smoosely\n";
107                 } else exit(0);
108             } break;
109             case 'a':{
110                 char filename[100];
111                 cout << "Enter a filename: ";
112                 cin.getline(filename, 99);
113                 char str[100];
114                 cout << "Enter text you want to write into your file: ";
115                 cin.getline(str, 99);
116                 if(write_to_file(filename, str, true)) {
117                     cout << "Writing file went smoosely\n";
118                 } else exit(1);
119             } break;
120             case 'c':{
121                 char copyF[100];
122                 char copyT[100];
123                 cout << "What file do you want to copy data from? ";
124                 cin >> copyF;
125                 cout << "What file do you want to copy data into? ";
126                 cin >> copyT;
127                 if(copy_file(copyF, copyT)) {
128                     cout << "Copying went smoosely\n";
129                 } else exit(2);
130             } break;
131             default:
132                 cout << "Incorrect input, try again:\n"; break;
133         }
134     }
135 }

```

Результат:

```

c:\practice_work_task_1_yevhen_kuteimakh } ; if ($?) { .\practice_work_task_1_yevhen_kuteimakh }
Hello, what do you want to do today?
But firstly you have to create a file
Enter a filename: test1
Enter text you want to write into your file: qweruyt
Writing file went smoosely
Now you can either add new string into your file or new file, or copy-paste from one file to another
Enter m for creating new file, a for adding new strings, c for copying, e for exit: a
Enter a filename: test1
Enter text you want to write into your file: ;hrehuiaouihsgfd
Writing file went smoosely
Enter m for creating new file, a for adding new strings, c for copying, e for exit: m
Enter a filename: test2
Enter text you want to write into your file: uasfgdihofguh
Writing file went smoosely
Enter m for creating new file, a for adding new strings, c for copying, e for exit: c
What file do you want to copy data from? test1
What file do you want to copy data into? testcopy
Copying went smoosely
Enter m for creating new file, a for adding new strings, c for copying, e for exit: r
Incorrect input, try again:
Enter m for creating new file, a for adding new strings, c for copying, e for exit: e
PS C:\Users\kutel\.vscode\projects\uni\epic5>

```

test1, test2, testcopy:

```

≡ test1  X
projects > uni > epic5 > ≡ test1
1  qweruyt
2  ;hrehuiaouihsgfd
3

```

```

≡ test2  X
projects > uni > epic5 > ≡ test2
1  uasfgdihofguh
2

```

```

≡ testcopy  X
projects > uni > epic5 > ≡ testcopy
1  qweruyt
2  ;hrehuiaouihsgfd
3

```

## Task 9 - Practice# programming: Self Practice Task

Назва для покемона

<https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/20048>

| Створено      | Компілятор | Результат               | Час (сек.) | Пам'ять (МіБ) | Дії                      |
|---------------|------------|-------------------------|------------|---------------|--------------------------|
| 16 годин тому | C++ 23     | Зарховано               | 0.003      | 1.188         | <a href="#">Перегляд</a> |
| 16 годин тому | C++ 23     | Неправильна відповідь 1 | 0.003      | 0.914         | <a href="#">Перегляд</a> |

Я обрав цю задачу оскільки вона полягає в роботі зі стрічками, що і було однією з головних тем епіку.

Умова:

### Назва для покемона

*Обмеження: 2 сек., 256 МБ*

Тарас знайшов новий вид покемонів, і тепер йому залишилося тільки придумати назву для нього. Ваше завдання — допомогти йому в цьому.

Наразі у Тараса є рядок  $t$ , який складається із символів англійського алфавіту, а також знаків запитання. Для того, щоб завершити процес придумування назви, потрібно кожен знак запитання замінити на довільний символ.

Крім цього відомо, що новий вид споріднений з іншим, давно відкритим видом під назвою  $s$ . Оскільки Тарас хоче, щоб назва нового виду відображала цю спорідненість, вона має містити  $s$  як підрядок.

Визначте, чи може Тарас замінити знаки запитання в назві  $t$  на символи таким чином, щоб відобразити спорідненість двох покемонів у назві.

#### Вхідні дані

У першому та другому рядках задано два рядки  $t$  та  $s$ .

Гарантується, що перший рядок складається з великих латинських літер та знаків запитання, а другий — лише з великих латинських літер.

#### Вихідні дані

В єдиному рядку виведіть **YES**, якщо Тарас може придумати назву для нового покемона, або **NO** в протилежному разі.

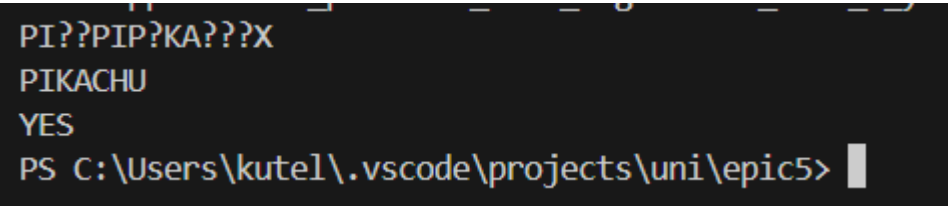
Задача не дуже складна, тому я вирішив її за 15 хв.

Програма:

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  bool check(string s1, string s2, int index) {
6      bool b = true;
7      for(int i = 1; i < s2.size(); i++) {
8          if(s1[i+index] == '?')
9              continue;
10         if(s1[i+index]!=s2[i]) {
11             b = false;
12             break;
13         }
14     }
15     return b;
16 }
17
18 int main() {
19     string t;
20     getline(cin, t);
21     string s;
22     getline(cin, s);
23     int lenStart = t.size();
24     int lenRes = s.size();
25     bool d = false;
26     for(int i = 0; i <= lenStart - lenRes; i++) {
27         if(t[i]==s[0] || t[i]=='?') {
28             if(check(t, s, i)) {
29                 d = true;
30                 break;
31             }
32         }
33     }
34     if(d) cout << "YES";
35     else cout << "NO";
36     return 0;
37 }
```

Результат:



## Робота у команді

У команді ми обговорювали поставлені перед нами цілі 5-го епіку. Колеги допомогли мені зрозуміти як відбувається взаємодія з файлами на мові C++, також ми розібрали практичне завдання та методи його реалізації. Ось скріншот однієї з наших зустрічей:



**Висновок:** Я навчився працювати з файлами у мові C++, зчитувати з/в них дані, отримав практичні навички роботи з файлами, та рядковими змінними. Дізнався, що таке Стандартна бібліотека, навчився нею користуватись. Навчився створювати власні бібліотеки та працювати з ними.