

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



## Звіт

### **про виконання лабораторних та практичних робіт блоку № 3**

На тему: «Цикли. Вкладені Цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функцій. Функції з змінною кількістю параметрів (еліпсис).

Рекурсія. Вбудовані функції.»

**з дисципліни:** «Основи програмування»

до:

ВНС Лабораторної Роботи № 2

ВНС Лабораторної Роботи № 3

ВНС Лабораторної Роботи № 7

Практичних Робіт до блоку № 3

**Виконав:**

Студент групи ІІІ-12  
Іваник Тарас Юрійович

Львів 2024

## Тема роботи:

Цикли. Вкладені Цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функцій. Функції з змінною кількістю параметрів (еліпсис). Рекурсія. Вбудовані функції.»

## Мета роботи:

Навчитися писати основні функції, цикли в C++. Розібрати що таке простори імен та як їх використовувати. Що таке перевантажені функції та навіщо вони потрібні, навчитись будувати блок-схеми складнішого типу.

## Теоретичні відомості:

[Уроки Acode по C++](#)

[Tutorial by BroCode](#)

University lectures

## Виконання роботи:

### Варіант 8

Lab# programming: VNS Lab 2 (60 хв)

8) Знайти суму ряду з точністю  $\epsilon=0.0001$ , загальний член якого

$$a_n = \frac{(2n-1)}{2^n}$$

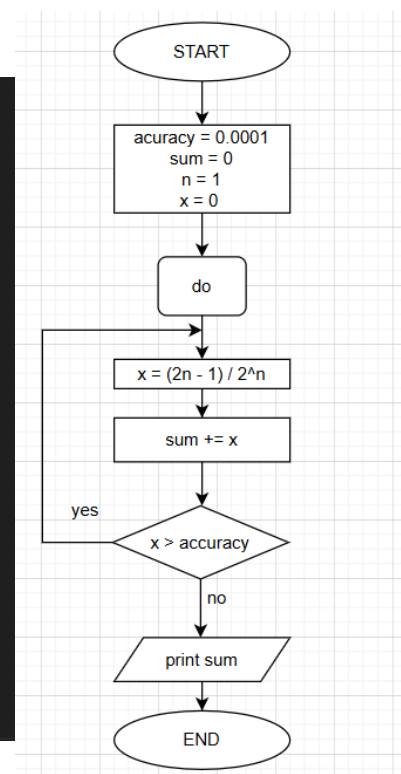
## Розв'язок:

## Результат виконання:

2.9999

PS C:\Users\User> █

```
1 #include <iostream>
2 #include <iomanip>
3 #include <cmath>
4
5 int main() {
6     double accuracy = 0.0001;
7     double sum = 0.0;
8     int n = 1;
9     double x;
10
11     do {
12         x = (2 * static_cast<double>(n) - 1) / pow(2, n);
13
14         if (std::abs(x) < accuracy)
15             break;
16
17         sum += x;
18         n++;
19     } while (true);
20
21     std::cout << std::fixed << std::setprecision(4) << sum << std::endl;
22
23     return 0;
24 }
```



## Lab# programming: VNS Lab 3(80 хв)

Для  $x$ , що змінюється від  $a$  до  $b$  з кроком  $(b-a)/k$ , де  $(k=10)$ , обчислити функцію  $f(x)$ , використовуючи її розклад в степеневий ряд у двох випадках:

а) для заданого  $n$ ;

б) для заданої точності  $\varepsilon$  ( $\varepsilon=0.0001$ ).

Для порівняння знайти точне значення функції.

8	$y = \frac{x \sin \frac{\pi}{4}}{1 - 2x \cos \frac{\pi}{4} + x^2}$	$0,1 \leq x \leq 0,8$	40	$S = x \sin \frac{\pi}{4} + x^2 \sin 2 \frac{\pi}{4} + \dots + x^n \sin n \frac{\pi}{4}$
---	--	-----------------------	----	--

```
#include <iostream>
#include <cmath>

using namespace std;
const double PI = 3.1415;

double exactValueOfx(double x){
    return (x*sin(PI/4)) / (1 - 2*x*cos(PI/4) + pow(x,2));
}

double sumForGivenN(double x, int n){
    double sum = 0.0;
    for(int k = 1; k <= n; k++){
        sum = pow(x,k) * sin(k) * PI/4 ;
    }
    return sum;
}

double sumPrecision(double x, double epsilon){
    double sum = 0.0;
    double a = 0.0;
    int k = 1;
    do{
        a = pow(x,k) * sin(k) * PI/4 ;
        sum += a;
        k++;
    } while (abs(a) > epsilon);
    return sum;
}

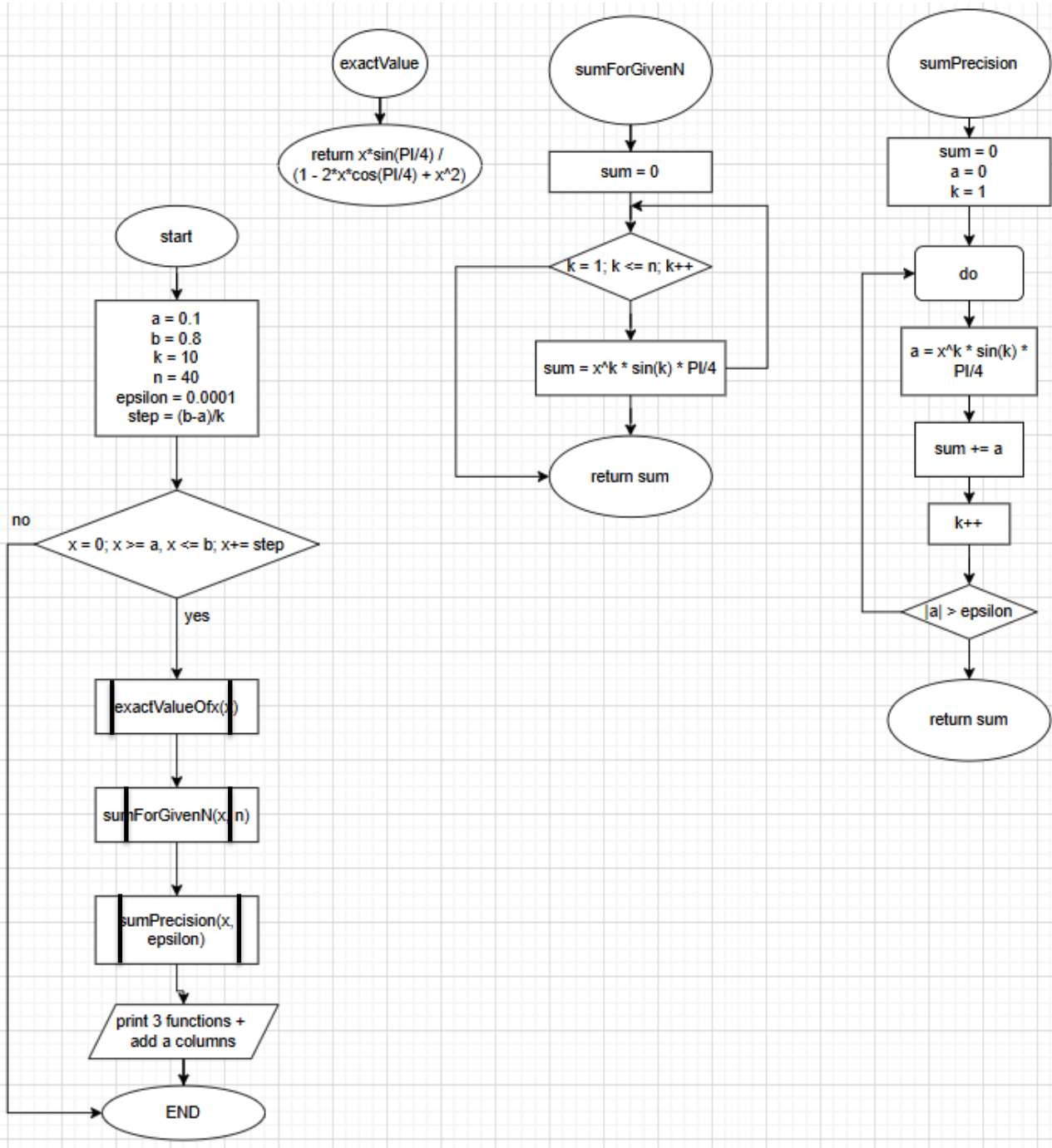
int main(){
    double a = 0.1;
    double b = 0.8;
    int k = 10;
    int n = 40;
    double epsilon = 0.0001;
    double step = (b - a) / k;

    for (double x = 0; x >= a, x <= b; x+= step){
        double exactValue = exactValueOfx(x);
        double sumForN = sumForGivenN(x, n);
        double precisionSum = sumPrecision(x, epsilon);
        cout << "X= " << x << "    SN= " << exactValue << "    SE= "<< sumForN << "    Y= " << precisionSum << endl;
    }
    return 0;
}
```

```

X= 0   SN= 0   SE= 0   Y= 1
X= 0.07 SN= 0.0546376 SE= 3.72581e-47 Y= 1
X= 0.14 SN= 0.120487 SE= 4.09657e-35 Y= 1
X= 0.21 SN= 0.198752 SE= 4.52972e-28 Y= 1
X= 0.28 SN= 0.290126 SE= 4.50423e-23 Y= 1
X= 0.35 SN= 0.394384 SE= 3.38861e-19 Y= 1
X= 0.42 SN= 0.509906 SE= 4.98048e-16 Y= 1
X= 0.49 SN= 0.63327 SE= 2.37215e-13 Y= 1
X= 0.56 SN= 0.759114 SE= 4.95245e-11 Y= 1
X= 0.63 SN= 0.8805 SE= 5.50708e-09 Y= 1
X= 0.7 SN= 0.989872 SE= 3.72581e-07 Y= 1
X= 0.77 SN= 1.08043 SE= 1.68627e-05 Y= 1
PS C:\Users\User>

```



## Lab# programming: VNS Lab 7 Task 1 (70 хв)

8. Написати функцію `max` зі змінною кількістю параметрів, що знаходить мінімальне із чисел типу `int`. Написати викликаючу функцію `main`, що звертається до функції `min` не менше трьох разів з кількістю параметрів 5, 10, 12.

```
#include <iostream>
#include <cstdarg>
#include <climits>

int min(int count, ...) {
    va_list args; //оголошення об'єкта для зберігання аргументів змінної довжини
    va_start(args, count);

    int minValue = INT_MAX;

    for (int i = 0; i < count; ++i) {
        int num = va_arg(args, int);
        if (num < minValue) {
            minValue = num;
        }
    }

    va_end(args);
    return minValue;
}

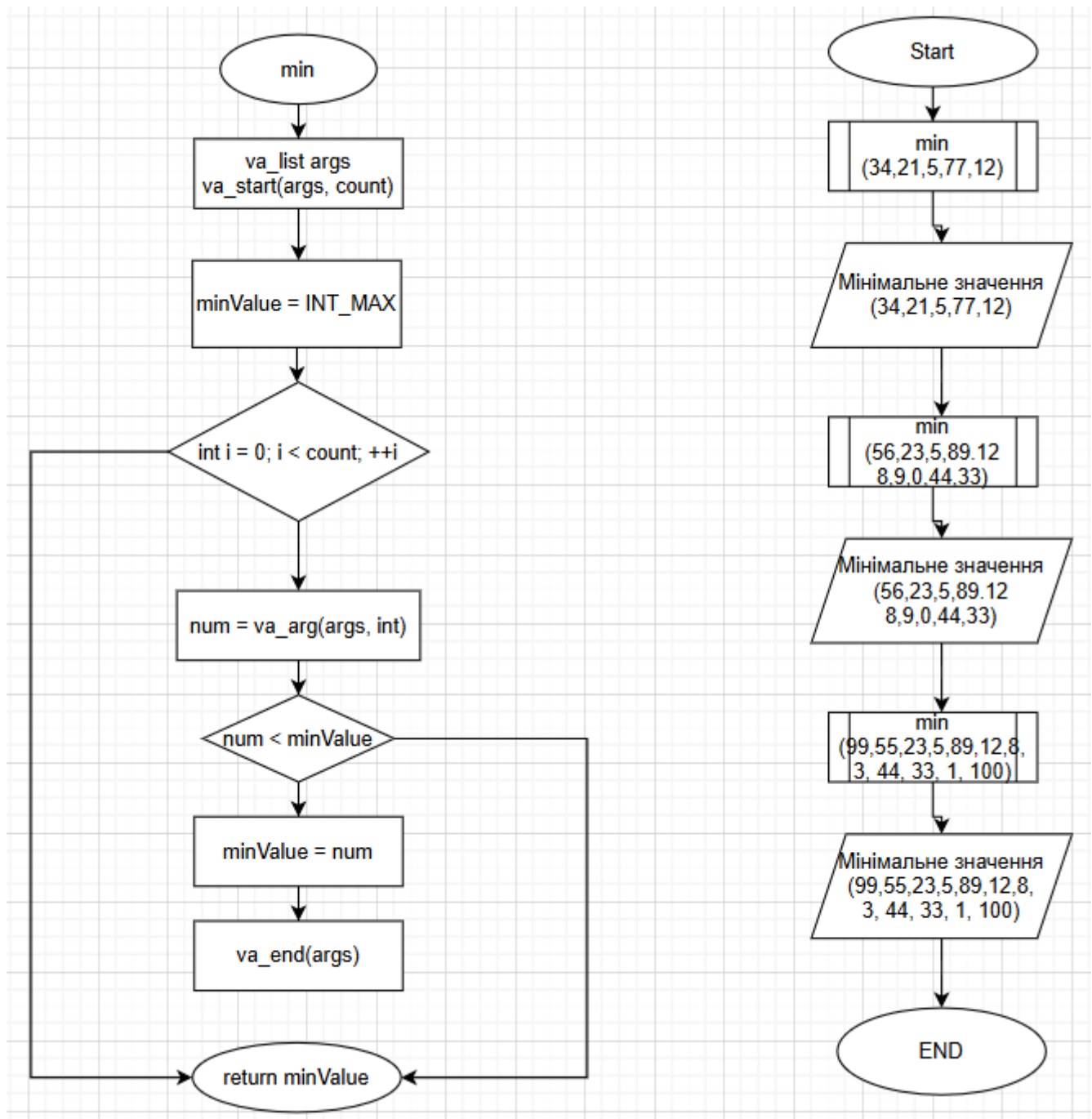
int main() {

    int result1 = min(5, 34, 21, 5, 77, 12);
    int result2 = min(10, 56, 23, 5, 89, 12, 8, 9, 0, 44, 33);
    int result3 = min(12, 99, 55, 23, 5, 89, 12, 8, 3, 44, 33, 1, 100);

    std::cout << "Мінімальне значення серед перших 5 чисел: " << result1 << std::endl;
    std::cout << "Мінімальне значення серед перших 10 чисел: " << result2 << std::endl;
    std::cout << "Мінімальне значення серед перших 12 чисел: " << result3 << std::endl;

    return 0;
}
```

```
Мінімальне значення серед перших 5 чисел: 5
Мінімальне значення серед перших 10 чисел: 0
Мінімальне значення серед перших 12 чисел: 1
PS C:\Users\User>
```



### Lab# programming: VNS Lab 7 Task 2 (50 хв)

2) Написати перевантажені функції й основну програму, що їх викликає.

- а) за номером року видає його назву за старояпонським календарем;
- б) за назвою місяця видає знак Зодіаку.

```

#include <iostream>
#include <string>

std::string get(int year) {
    const std::string years[] = {
        "Щур", "Бик", "Тигр", "Кролик", "Дракон", "Змія",
        "Кінь", "Коза", "Мавпа", "Півень", "Собака", "Кабан"
    };
    int index = (year - 4) % 12;
    return years[index];
}

std::string get(const std::string& month) {
    if (month == "January" || month == "February") return "Водолій";
    if (month == "February" || month == "March") return "Риби";
    if (month == "March" || month == "April") return "Овен";
    if (month == "April" || month == "May") return "Телець";
    if (month == "May" || month == "June") return "Близнюки";
    if (month == "June" || month == "July") return "Рак";
    if (month == "July" || month == "August") return "Лев";
    if (month == "August" || month == "September") return "Діва";
    if (month == "September" || month == "October") return "Терези";
    if (month == "October" || month == "November") return "Скорпіон";
    if (month == "November" || month == "December") return "Стрілець";
    if (month == "December" || month == "January") return "Козеріг";

    return "Unknown month";
}

int main() {

    std::cout << "Enter a year: ";
    int year;
    std::cin >> year;
    std::cout << "Ваш рік за старояпонським календарем, це: " << get(year) << std::endl;

    std::cout << "Enter a month ";
    std::string month;
    std::cin >> month;
    std::cout << "Ваш знак зодіаку: " << get(month) << std::endl;

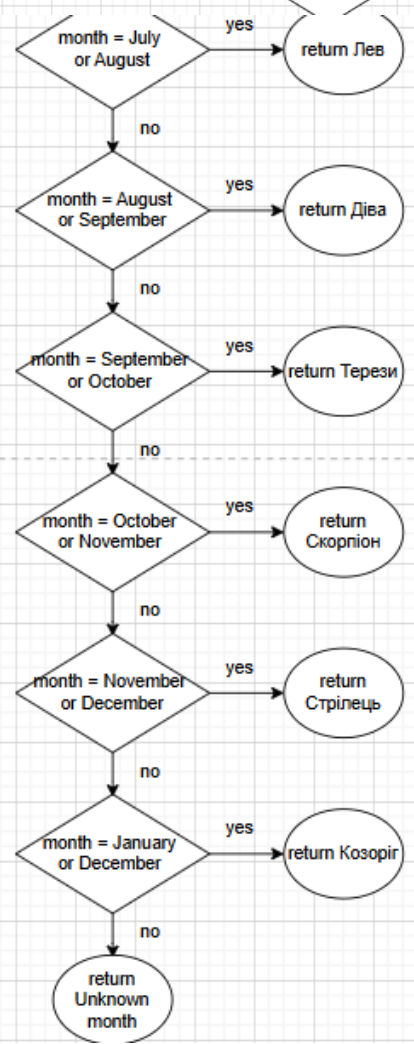
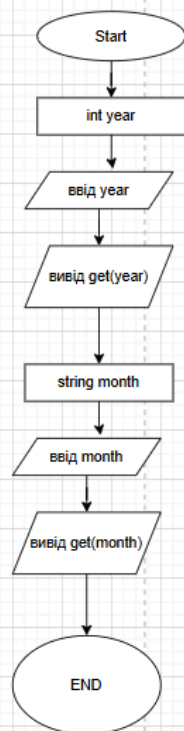
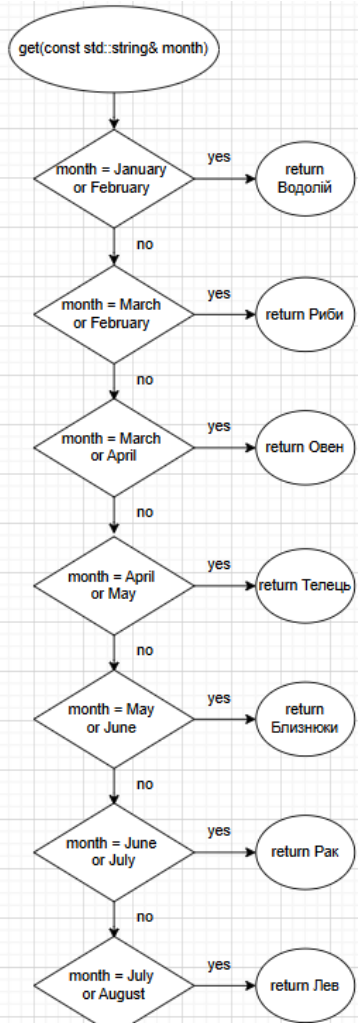
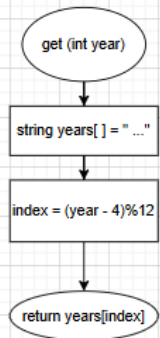
    return 0;
}

```

```

Enter a year: 2024
Ваш рік за старояпонським календарем, це: Дракон
Enter a month September
Ваш знак зодіаку: Діва
PS C:\Users\User>

```





## Practice# programming: Class Practice Task [Library] (270 xe)

```
#include <iostream>
#include <vector>
#include <string>

using namespace std;

void showMenu() {
    cout << "***** MENU *****" << endl;
    cout << "1. Enumerate all books" << endl;
    cout << "2. Borrow a book" << endl;
    cout << "3. Return a book" << endl;
    cout << "4. EXIT" << endl;
}

void listBooks(const vector<string>& books, const vector<bool>& available) {
    cout << "List of books: " << endl;
    for (size_t i = 0; i < books.size(); ++i) {
        cout << i + 1 << ". " << books[i] << (available[i] ? " (available)" : " (not available)") << endl;
    }
}

void borrowBook(vector<bool>& available) {
    cout << "Enter a number of the book you want to borrow: " << endl;
    int bookNumber;
    cin >> bookNumber;

    if (bookNumber < 1 || bookNumber > available.size() || !available[bookNumber - 1]) {
        cout << "Book is not available or you entered the wrong number." << endl;
        return;
    }

    available[bookNumber - 1] = false;
    cout << "You borrowed a book." << endl;
}

void returnBook(vector<bool>& available) {
    cout << "Enter a number of the book you want to return: " << endl;
    int bookNumber;
    cin >> bookNumber;

    if (bookNumber < 1 || bookNumber > available.size() || available[bookNumber - 1]) {
        cout << "Book wasn't borrowed or you entered a wrong number." << endl;
        return;
    }
}
```

```

int main() {
    vector<bool> available = {true, true, true, true, true};
    vector<string> books = {"Harry Potter", "Sword of King Arthur", "Gentlemen", "El Camino", "Bad Boys"};

    int choice;

    while (true) {
        showMenu();
        cout << "Your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                listBooks(books, available);
                break;
            case 2:
                borrowBook(available);
                break;
            case 3:
                returnBook(available);
                break;
            case 4:
                cout << "Thank you, for using our library, see you" << endl;
                return 0;
            default:
                cout << "Incorrect choice, try again!" << endl;
                continue;
        }

        cout << "Do you want to do another operation? (1 - yes, 0 - no): ";
        int continueChoice;
        cin >> continueChoice;

        if (continueChoice != 1) {
            break;
        }
    }

    return 0;
}

```

```

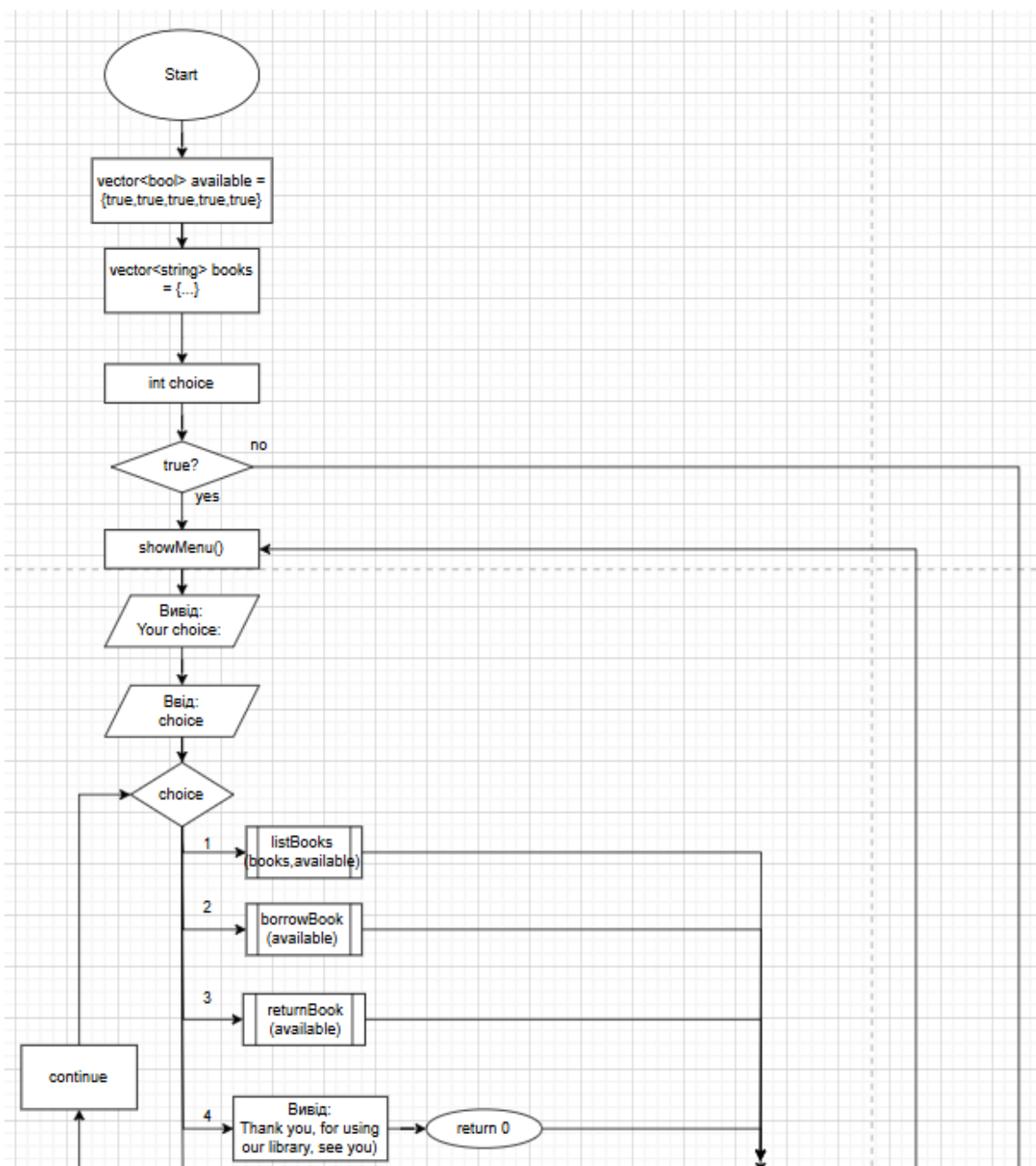
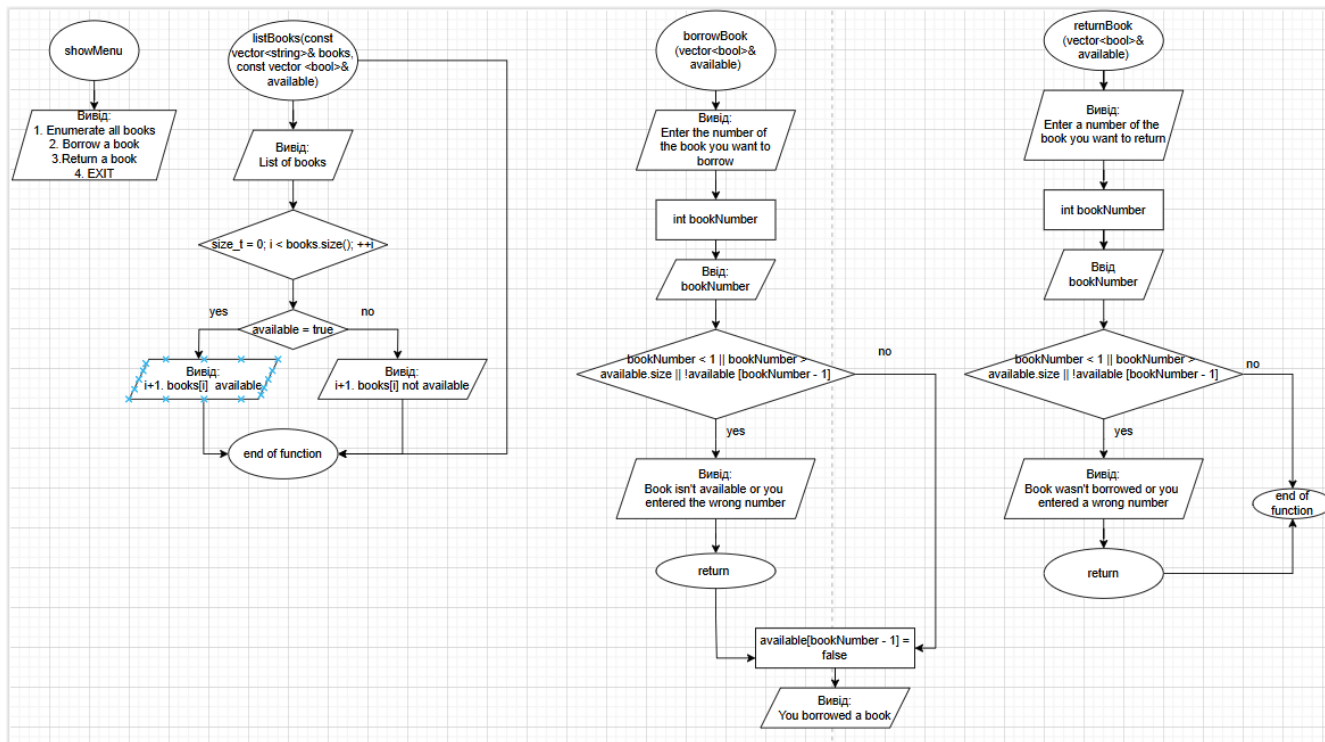
***** MENU *****
1. Enumerate all books
2. Borrow a book
3. Return a book
4. EXIT
Your choice: 1
List of books:
1. Harry Potter (available)
2. Sword of King Arthur (available)
3. Gentlemen (available)
4. El Camino (available)
5. Bad Boys (available)
Do you want to do another operation? (1 - yes, 0 - no): 1
***** MENU *****
1. Enumerate all books
2. Borrow a book
3. Return a book
4. EXIT
Your choice: 2
Enter a number of the book you want to borrow:
2
You borrowed a book.
Do you want to do another operation? (1 - yes, 0 - no): 1
***** MENU *****
1. Enumerate all books
2. Borrow a book
3. Return a book
4. EXIT
Your choice: 1
List of books:
1. Harry Potter (available)
2. Sword of King Arthur (not available)
3. Gentlemen (available)
4. El Camino (available)
5. Bad Boys (available)
Do you want to do another operation? (1 - yes, 0 - no): 1

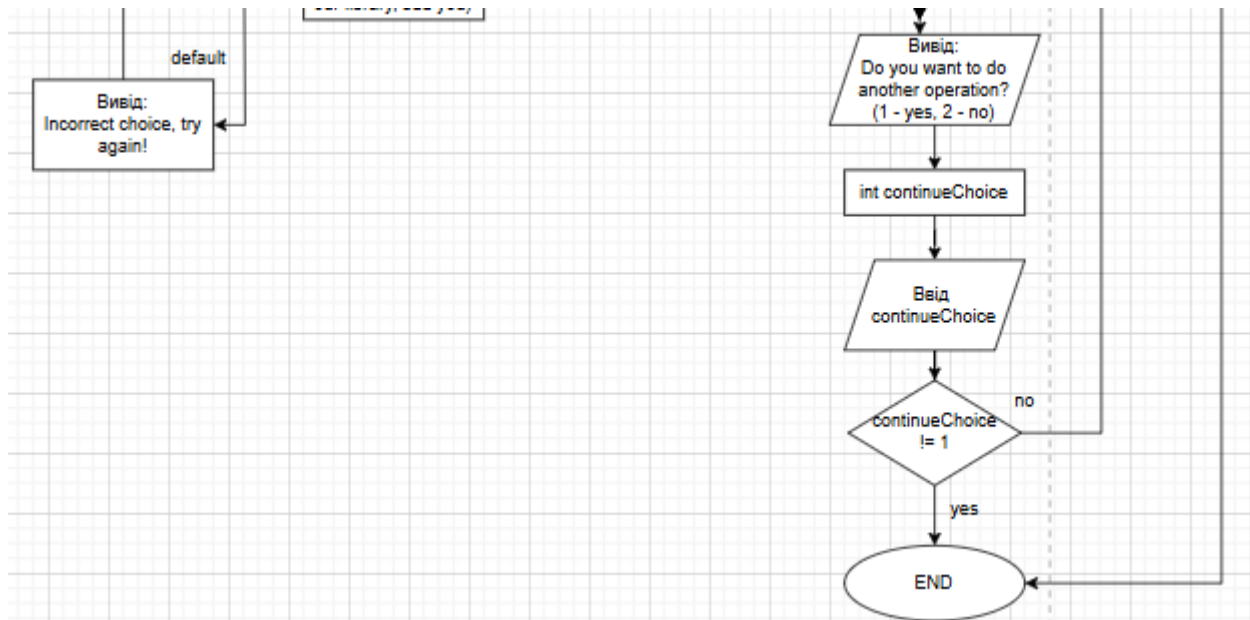
```

```

Do you want to do another operation? (1 - yes, 0 - no): 1
***** MENU *****
1. Enumerate all books
2. Borrow a book
3. Return a book
4. EXIT
Your choice: 3
Enter a number of the book you want to return:
2
You returned a book.
Do you want to do another operation? (1 - yes, 0 - no): 1
***** MENU *****
1. Enumerate all books
2. Borrow a book
3. Return a book
4. EXIT
Your choice: 1
List of books:
1. Harry Potter (available)
2. Sword of King Arthur (available)
3. Gentlemen (available)
4. El Camino (available)
5. Bad Boys (available)
Do you want to do another operation? (1 - yes, 0 - no): 1
***** MENU *****
1. Enumerate all books
2. Borrow a book
3. Return a book
4. EXIT
Your choice: 4
Thank you, for using our library, see you)

```





## Practice# programming: Self Practice Algotester Task(70 хв)

### Гра в карти

Limits: 2 sec., 256 MiB

Нещодавно ми мали змогу спостерігати за рідкісним явищем. Блакитний кривавий супермісяць — однозначно незабутнє видовище. Звісно ж, наш давній знайомий романтик Зенік не міг пропустити таку нагоду вразити Марічку. Тож він запросив її додому й розповідав про дивовижний сюрприз, який чекає на неї вночі.

Проте Марічка вперто не хотіла залишатися на ніч. І Зенік придумав, як її зацікавити. Він знайшов колоду з  $n$  карт. На кожній карті була записана мала літера латинського алфавіту. Завдання Марічки — скласти з наявних карт найдовший можливий паліндром.

**Паліндром** — це рядок, який однаково читається зліва направо та справа наліво. Наприклад, рядки `a`, `aba`, `pwpp` є паліндромами, водночас стрічки `ab`, `abc`, `aabb` — ні.

### Input

У першому рядку задано ціле число  $n$  — кількість карт.

У другому рядку задано  $n$  малих літер латинського алфавіту  $c_i$  — літера, написана на  $i$ -ій карті.

### Output

Виведіть один рядок — найдовший паліндром, який можна скласти з літер, написаних на картах. Якщо таких рядків декілька, виведіть будь-який.

### Constraints

$1 \leq n \leq 10^3$ .

### Samples

Input (stdin)	Output (stdout)
7 a b c c z b w	bczcb

- About
- Statistics
- Help
- Analysis
- Events
- TARAS NANYK
- My Solutions
- My teams
- ARCHIVE
- Problems
- Rank
- Solution Queue

Submit a solution

File

Source Code

Compiler

C++ 23

File

Вибрати файл

Файл не вибрано

Submit

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
a few seconds ago	C++ 23	Accepted	0.003	1.293	<a href="#">View</a>
4 hours ago	C++ 23	Accepted	0.002	1.625	<a href="#">View</a>
4 hours ago	C++ 23	Accepted	0.003	1.262	<a href="#">View</a>
4 hours ago	C++ 23	Wrong Answer 1	0.003	0.914	<a href="#">View</a>
4 hours ago	C++ 23	Wrong Answer 1	0.002	0.941	<a href="#">View</a>
4 hours ago	C++ 23	Wrong Answer 1	0.002	0.914	<a href="#">View</a>
4 hours ago	C++ 23	Wrong Answer 1	0.002	0.910	<a href="#">View</a>

Showing 1 to 7 of 7 rows

Гра в карти: Submission #1858824

Accepted

```

#include <iostream>
#include <map>
#include <string>
#include <sstream> //for stringstream (дає змогу обробляти рядок і вибирати з нього окремі букви)
#include <vector>

using namespace std;

int main() {
    int n;
    cin >> n;
    cin.ignore();

    string line;
    getline(cin, line);

    stringstream ss(line); //stringstream проходить по кожній букві окремо
    vector<char> cards;
    string letter;

    while (ss >> letter) { // [ss >> letter] записує кожне слово з рядка в змінну letter
        if (letter.length() == 1 && isalpha(letter[0])) {
            cards.push_back(tolower(letter[0])); // Додаємо літери в нижньому регістрі
        }
    }

    if (cards.size() > n) {
        cout << "Error: Too many letters entered!" << endl;
        return 1;
    }

    map<char, int> freq; // мап для алфавітного порядку
    for (char c : cards) {
        freq[c]++; // збільшуємо лічильник кожен раз як зустрічається одна й та сама буква
    }

    string half = "", middle = "";

    for (auto &pair : freq) {
        char letter = pair.first;
        int count = pair.second;

        half += string(count / 2, letter);

        if (count % 2 != 0 && middle.empty()) { //якщо непарна к-сть, то додаємо символ для середини
            middle = letter;
        }
    }

    string palindrome = half + middle + string(half.rbegin(), half.rend());
    cout << palindrome << endl;

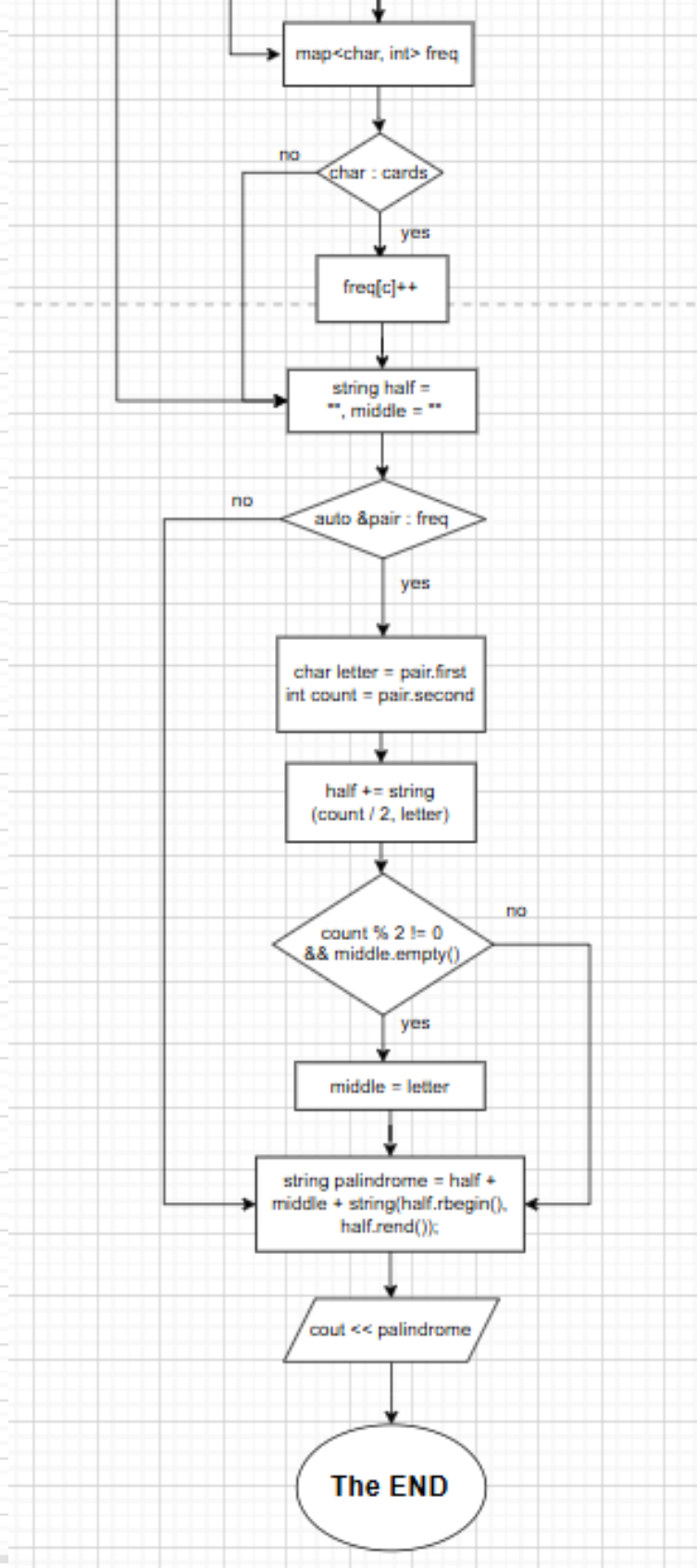
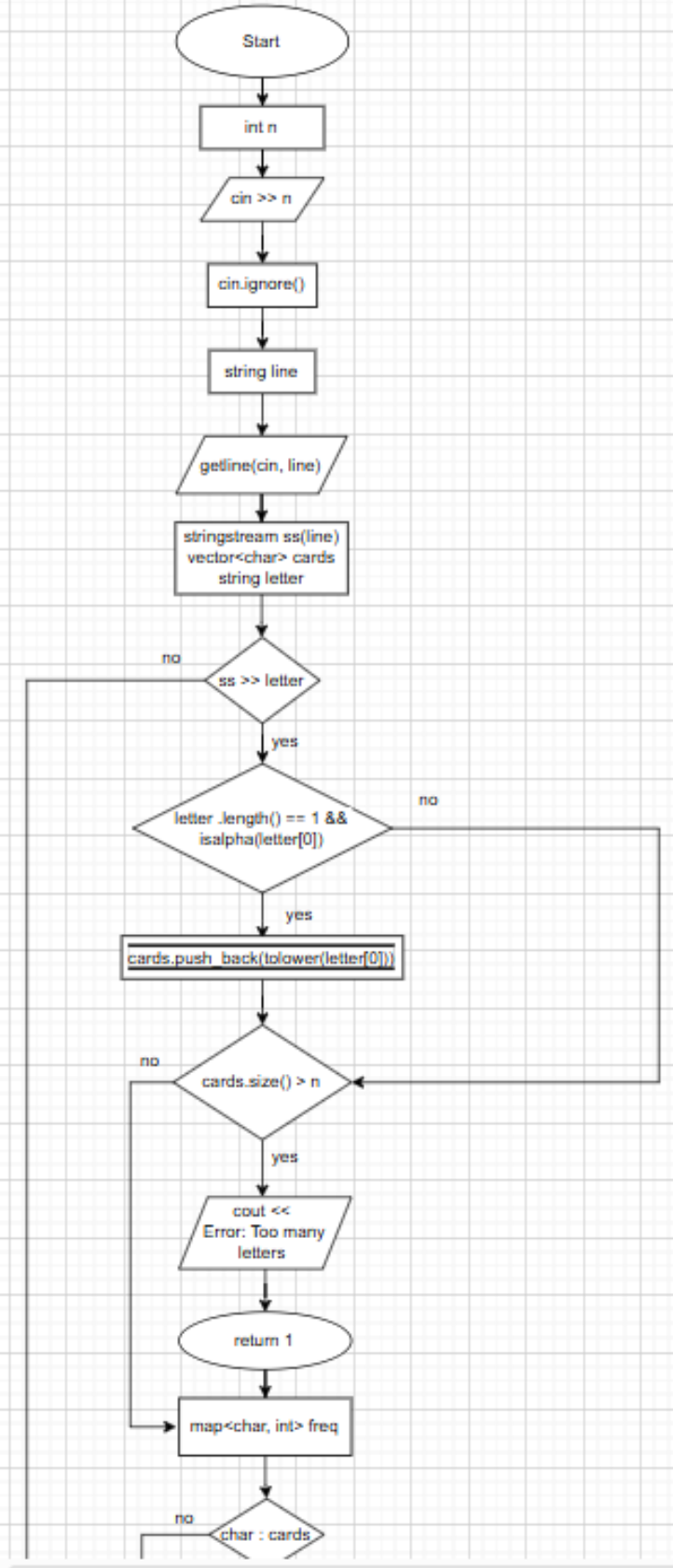
    return 0;
}

```

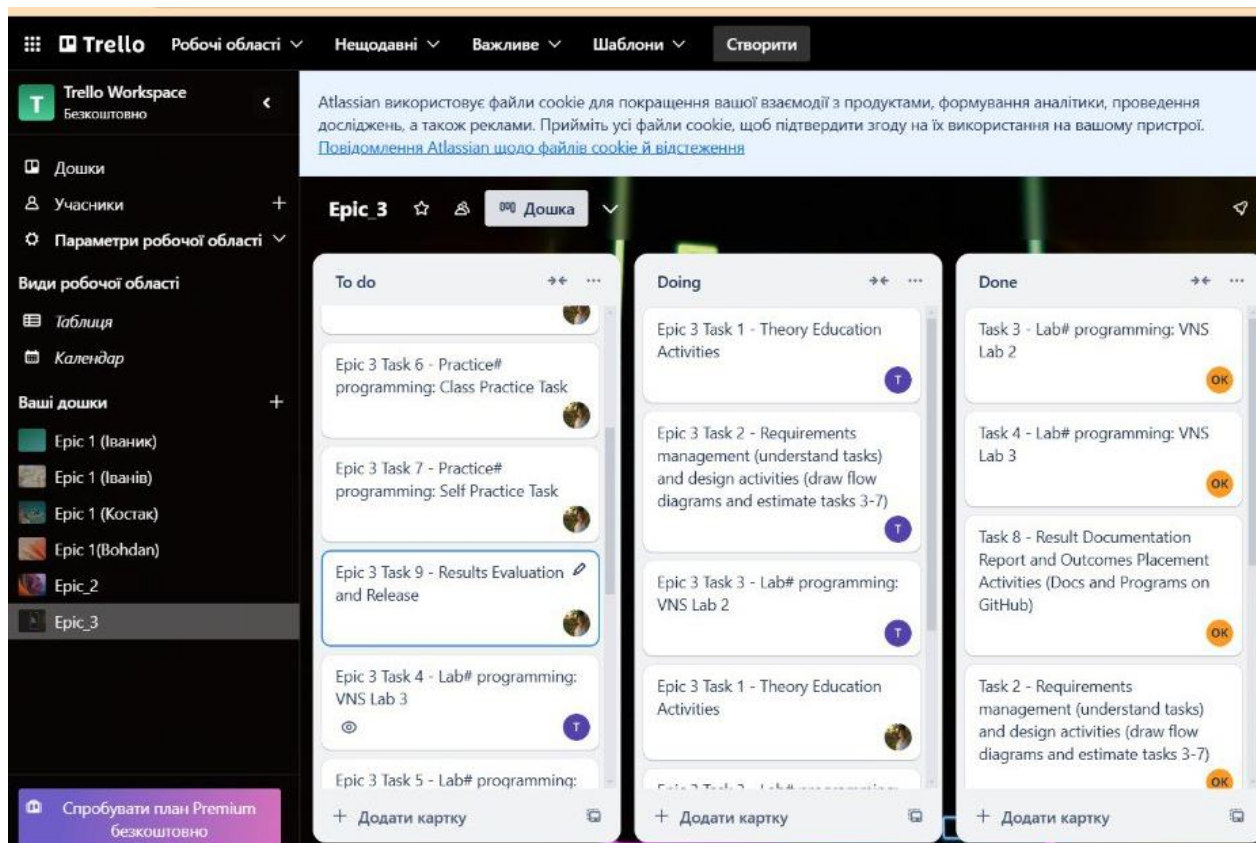
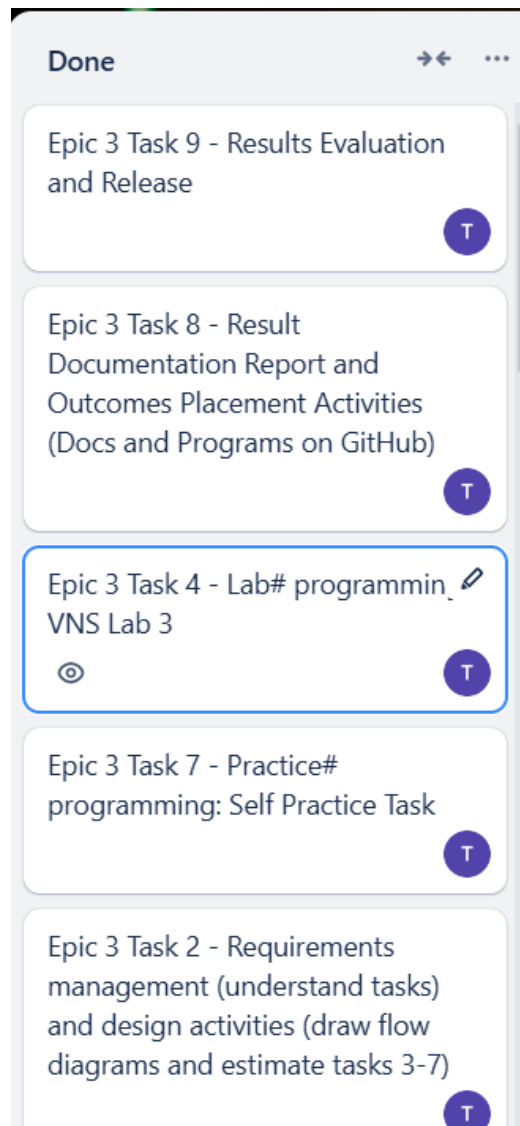
```

7
a b c c z b w
bcacb
PS C:\Users\User>

```



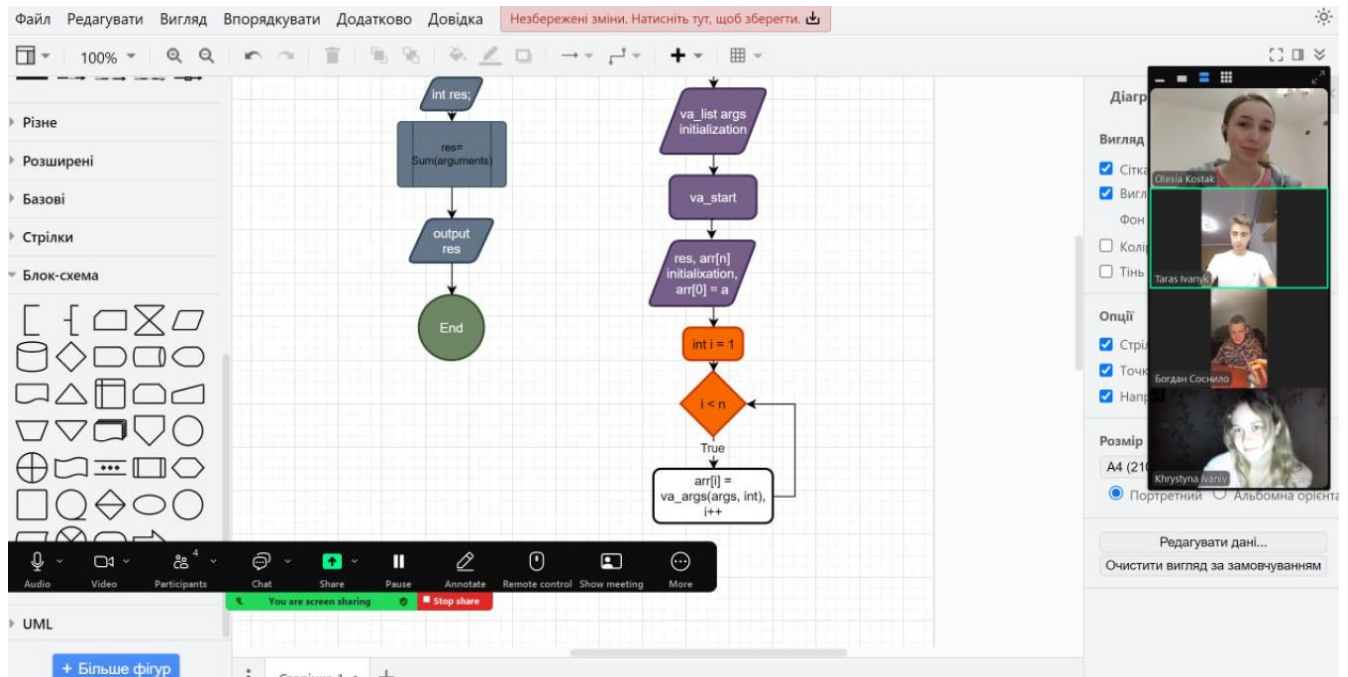
# TRELLO:





## Pull request

**Висновок:** я навчився робити складні блок-схеми в draw.io, також використовувати різні типи циклів і керувати ними, навчився юзати перевантажені функції і простори імен, і багато чого іншого.



**Була одна зустріч онлайн з командою по 3 Епіку.**