

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 ШІ-13

Шийка Стефан Андрійович

**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 використовувати функції різних видів, цикли та виконувати завдання над ними. Також будувати блок-схеми складніших типів.

### Теоретичні відомості:

- цикл for
- цикл while
- цикл do, while
- функції

Джерела:

- [aCode – перезавантаження функцій в C++](#)
- [aCode – еліпсис](#)
- [aCode – goto](#)
- [Harvard CS50 – теорія та задачі по циклах та рекурсії](#)

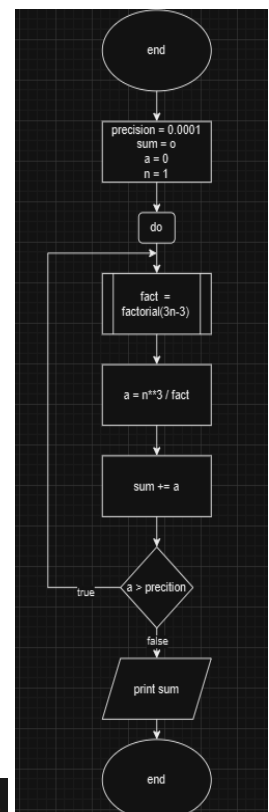
## Виконання роботи

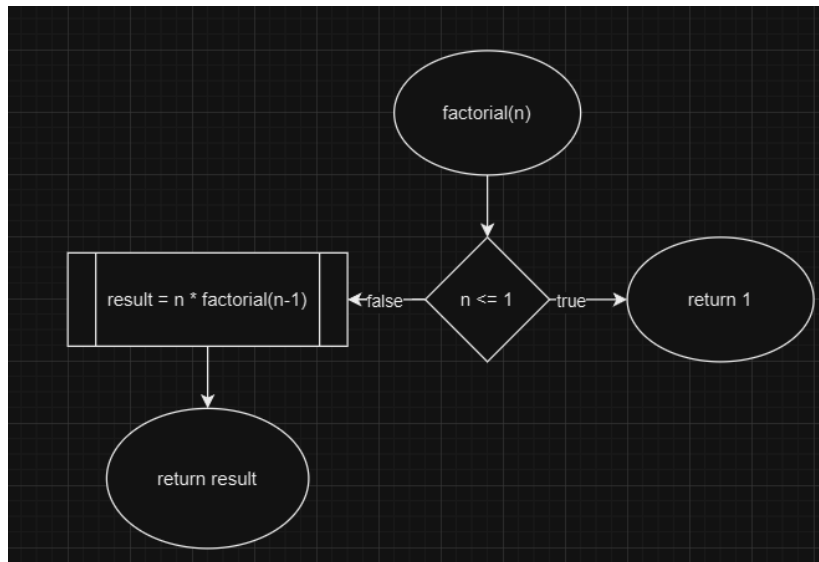
Варіант 18

### Lab# programming: VNS Lab 2

```
1  ✓ #include <iostream>
2  ✓ #include <math.h>
3
4  ✓ using namespace std;
5
6  ✓ unsigned long long factorial(int n){
7      if(n <= 1) return 1;
8
9      return n * factorial(n - 1);
10 }
11
12 ✓ int main(){
13     double precision = 0.0001;
14     double sum = 0;
15     double a = 0;
16     int n = 1;
17
18     do{
19         a = pow(n, 3) / (double)factorial(3 * n - 3);
20         sum += a;
21         n++;
22     }while(a > precision);
23
24     cout << sum << endl;
25 }
```

2.37101





18) Знайти суму ряду з точністю  $\varepsilon=0.0001$ , загальний член якого

$$a_n = \frac{n^3}{(3n-3)!}$$

### Lab# programming: VNS Lab 3

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:

- для заданого  $n$ ;
- для заданої точності  $\varepsilon$  ( $\varepsilon=0.0001$ ).

Для порівняння знайти точне значення функції.

|    |  |                       |    |  |
|----|--|-----------------------|----|--|
| 18 | $y = \frac{1}{2} - \frac{\pi}{4}  \sin x $ | $0,1 \leq x \leq 0,8$ | 50 | $S = \frac{\cos 2x}{3} + \frac{\cos 4x}{15} + \dots + \frac{\cos 2nx}{4n^2 - 1}$ |
|----|--|-----------------------|----|--|

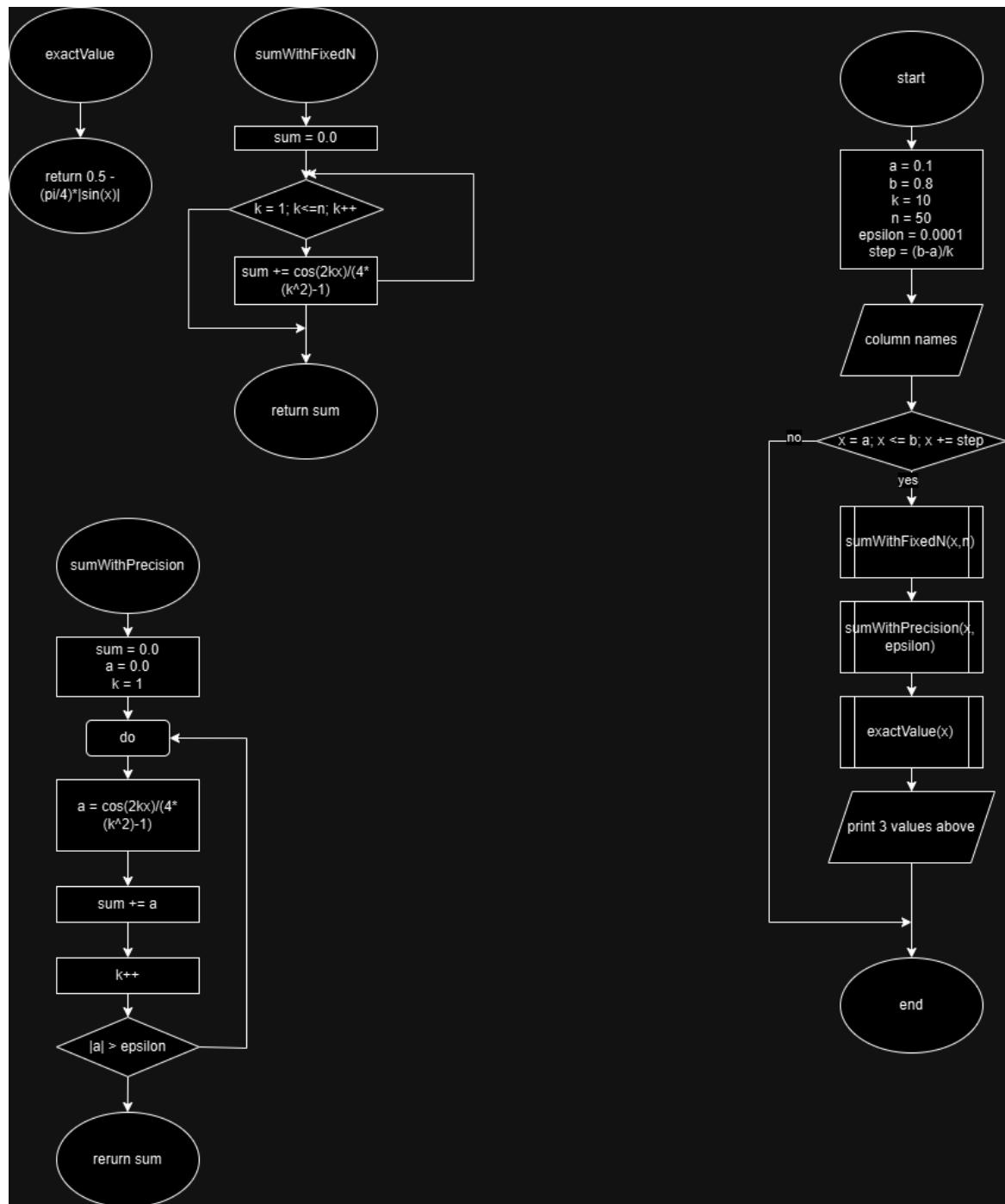
```

1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  double exactValue(double x) {
7      return 0.5 - (M_PI / 4) * fabs(sin(x));
8  }
9
10 double sumWithFixedN(double x, int n) {
11     double sum = 0.0;
12     for (int k = 1; k <= n; k++) {
13         sum += cos(2 * k * x) / (4 * k * k - 1);
14     }
15     return sum;
16 }
17
18 double sumWithPrecision(double x, double epsilon) {
19     double sum = 0.0;
20     double a = 0.0;
21     int k = 1;
22
23     do {
24         a = cos(2 * k * x) / (4 * k * k - 1);
25         sum += a;
26         k++;
27     } while (fabs(a) > epsilon);
28
29     return sum;
30 }
31
32 int main() {
33     double a = 0.1;
34     double b = 0.8;
35     int k = 10;
36     int n = 50;
37     double epsilon = 0.0001;
38
39     double step = (b - a) / k;
40
41     cout << "X\t\tsumWithN\t\tsumWithEpsilon\t\tfuncValue" << endl;
42
43     for (double x = a; x <= b; x += step) {
44         double sumWithN = sumWithFixedN(x, n);
45         double sumWithEpsilon = sumWithPrecision(x, epsilon);
46         double funcValue = exactValue(x);
47
48         cout << x << "\t\t" << sumWithN << "\t\t" << sumWithEpsilon << "\t\t" << funcValue << endl;
49     }
50 }
51

```

| X    | sumWithN   | sumWithEpsilon | funcValue  |
|------|------------|----------------|------------|
| 0.1  | 0.421368   | 0.419708       | 0.421591   |
| 0.17 | 0.366845   | 0.363998       | 0.367124   |
| 0.24 | 0.313139   | 0.312389       | 0.313309   |
| 0.31 | 0.260381   | 0.261103       | 0.260408   |
| 0.38 | 0.208758   | 0.208144       | 0.20868    |
| 0.45 | 0.15849    | 0.157981       | 0.158379   |
| 0.52 | 0.10983    | 0.109546       | 0.109751   |
| 0.59 | 0.0630465  | 0.0642786      | 0.0630351  |
| 0.66 | 0.0184101  | 0.0187129      | 0.0184592  |
| 0.73 | -0.0238311 | -0.0230671     | -0.0237582 |

PS C:\Users\user\ai\_programming\_playground\_2024\ai\_12\stefan\_shyika\Epic3>



## Lab# programming: VNS Lab 7

1)

18. Написати функцію `pok` зі змінною кількістю параметрів, що знаходить найменше спільне кратне для декількох чисел.

$$HCK(a, b) = \frac{a \cdot b}{HCD(a, b)}$$

(НСД- найбільший спільний дільник)

Написати викликаючу функцію `main`, що звертається до функції `pok` не менше трьох разів з кількістю параметрів 3, 5, 6.

```

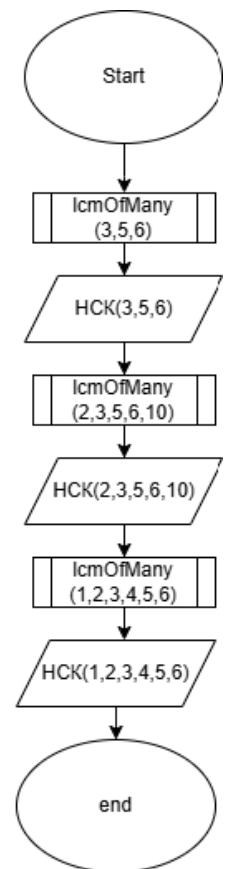
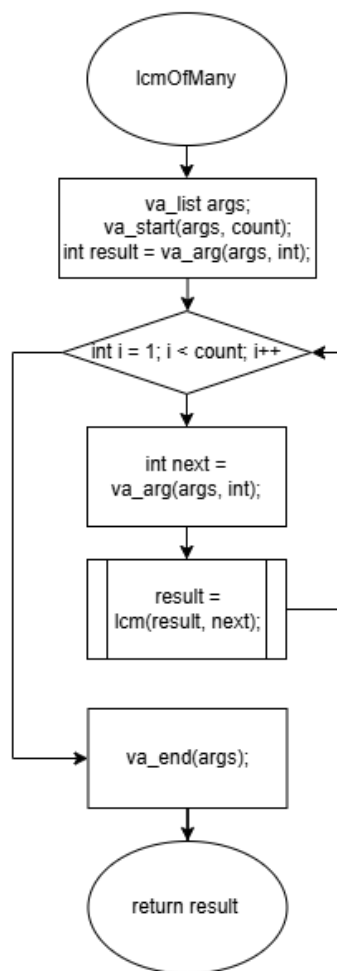
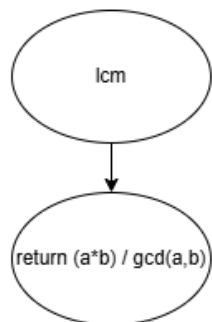
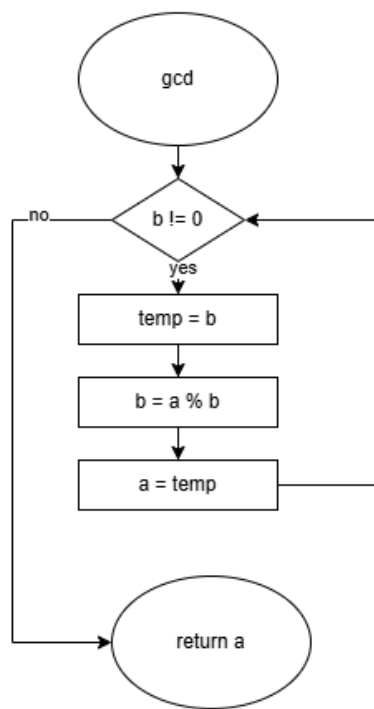
1  #include <iostream>
2  #include <cstdlib>
3  #include <algorithm>
4
5  using namespace std;
6
7  int gcd(int a, int b) {
8      while (b != 0) {
9          int temp = b;
10         b = a % b;
11         a = temp;
12     }
13     return a;
14 }
15
16 int lcm(int a, int b) {
17     return (a * b) / gcd(a, b);
18 }
19
20 int lcmOfMany(int count, ...) {
21     va_list args;
22     va_start(args, count);
23
24     int result = va_arg(args, int);
25
26     for (int i = 1; i < count; i++) {
27         int next = va_arg(args, int);
28         result = lcm(result, next);
29     }
30
31     va_end(args);
32     return result;
33 }
34
35 int main() {
36     cout << "HCK(3, 5, 6): " << lcmOfMany(3, 3, 5, 6) << endl;
37     cout << "HCK(2, 3, 5, 6, 10): " << lcmOfMany(5, 2, 3, 5, 6, 10) << endl;
38     cout << "HCK(1, 2, 3, 4, 5, 6): " << lcmOfMany(6, 1, 2, 3, 4, 5, 6) << endl;
39 }

```

HCK(3, 5, 6): 30

HCK(2, 3, 5, 6, 10): 30

HCK(1, 2, 3, 4, 5, 6): 60



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

18.

а) для масиву цілих чисел знаходить максимальний елемент;

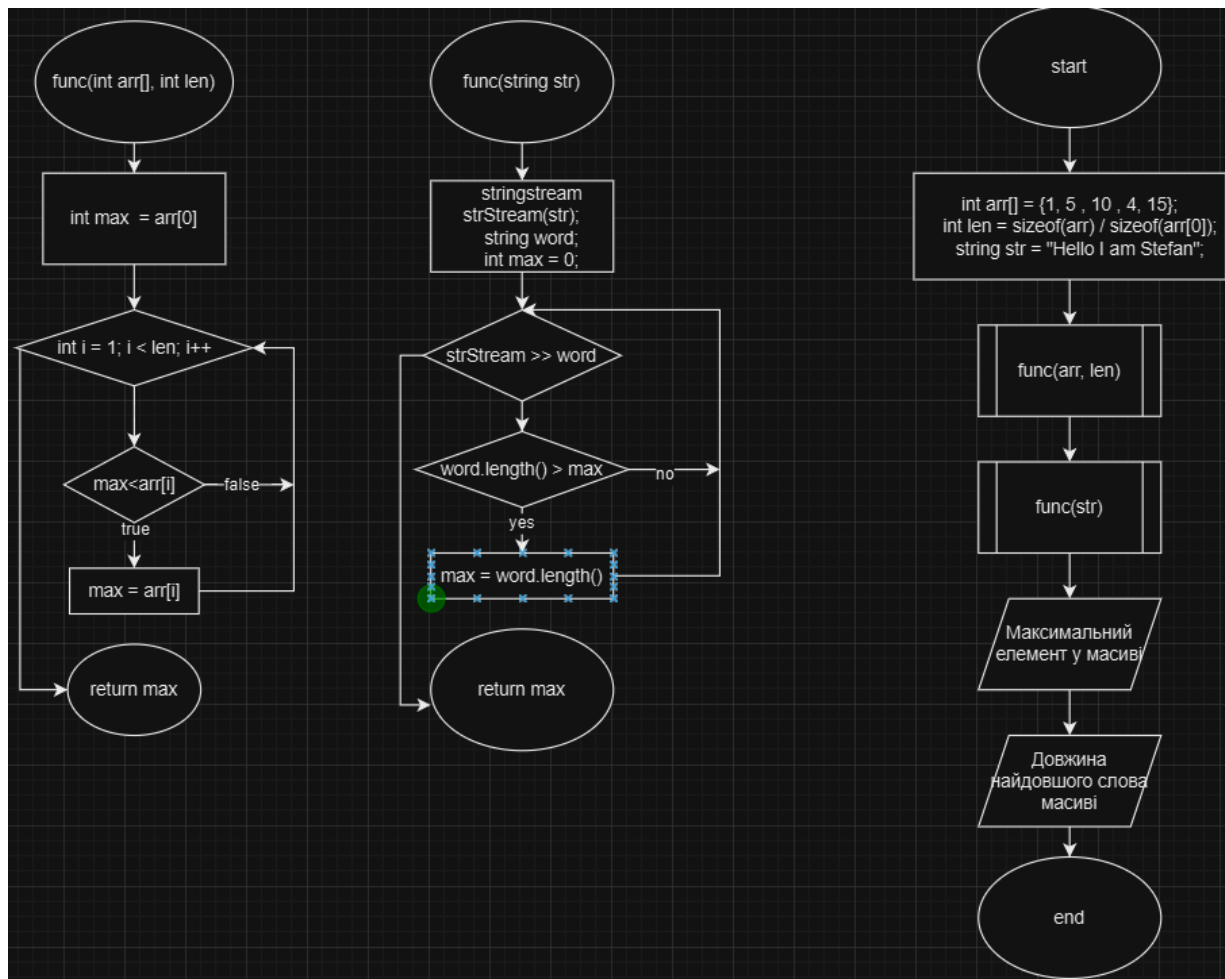
б) для рядка знаходить довжину найдовшого слова .

```
1  #include <iostream>
2  #include <string>
3  #include <cstring>
4  #include <sstream>
5
6  using namespace std;
7
8  int func(int arr[], int len){
9      int max = arr[0];
10
11     for(int i = 1; i < len; i++){
12         if(max < arr[i]){
13             max = arr[i];
14         }
15     }
16     return max;
17 }
18
19 int func(string str){
20     stringstream strStream(str);
21     string word;
22     int max = 0;
23
24     while (strStream >> word) {
25         if (word.length() > max) {
26             max = word.length();
27         }
28     }
29
30     return max;
31 }
32
33 int main(){
34     int arr[] = {1, 5 , 10 , 4, 15};
35     int len = sizeof(arr) / sizeof(arr[0]);
36     string str = "Hello I am Stefan";
37
38     cout << "Максимальний елемент у масиві: " << func(arr, len) << endl;
39
40     cout << "Довжина найдовшого слова: " << func(str) << endl;
41 }
```

Максимальний елемент у масиві: 15

Довжина найдовшого слова: 6





## Practice# programming: Class Practice Task

Бібліотека

```

1  #include <iostream>
2  #include <vector>
3  #include <ctype>
4
5  using namespace std;
6
7  void borrowBook();
8  void returnBook();
9  void listAllBooks();
10
11
12  vector<string> books = {"Book1", "Book2", "Book3", "Book4"};
13  vector<bool> isAvailable = {true, true, true, true};
14
15  int main(){
16      bool exit = false;
17
18      while (!exit) {
19          menu:
20
21          int choice;
22
23          cout << "\nМеню:\n";
24          cout << "1 Позичити книгу\n";
25          cout << "2 Повернути книгу\n";
26          cout << "3 Перерахувати всі книги\n";
27          cout << "4 Вийти\n";
28          cout << "Я хочу: ";
29
30          cin >> choice;
31
32          switch (choice) {
33              case 1:
34                  borrowBook();
35                  break;
36              case 2:
37                  returnBook();
38                  break;
39              case 3:
40                  listAllBooks();
41                  break;
42              case 4:
43                  exit = true;
44                  break;
45              default:
46                  cout << "Такої функції в нашій бібліотеці нема ;)\n";
47                  goto menu;
48          }
49
50          char anotherOperation;
51          do {
52              cout << "\nХочете виконати іншу операцію? (y/n): ";
53              cin >> anotherOperation;
54          } while (tolower(anotherOperation) != 'y' && tolower(anotherOperation) != 'n');
55
56          if (tolower(anotherOperation) == 'n') {
57              exit = true;
58          }
59      }
60  }
61
62  void listAllBooks(){
63      cout << "Список книг: \n";
64
65      for(int i = 0; i < books.size(); i++){
66          cout << i+1 << " " << books[i];
67          if(isAvailable[i]){
68              cout << "   В наявності\n";
69          }else{
70              cout << "   Не в наявності\n";
71          }
72      }
73  }
74
75  void borrowBook() {
76      int num;
77      listAllBooks();
78      cout << "\nВведіть номер книги, яку бажаєте взяти: ";
79      cin >> num;
80
81      if (num < 1 || num > books.size()) {
82          cout << "Такої книги нема.\n";
83          goto menu;
84      }
85
86      if (isAvailable[num - 1]) {
87          isAvailable[num - 1] = false;
88          cout << "Ви взяли книгу: " << books[num - 1] << "\n";
89      } else {
90          cout << "Книги нема наявності.\n";
91      }
92
93      menu:
94
95      return;
96  }
97
98  void returnBook(){
99      listAllBooks();
100      int num;
101      cout << "\nВведіть номер книги, яку бажаєте повернути: ";
102      cin >> num;
103
104      if (num < 1 || num > books.size()) {
105          cout << "Такої книги нема.\n";
106          goto menu;
107      }
108
109      if(!isAvailable[num - 1]) {
110          isAvailable[num - 1] = true;
111          cout << "Ви повернули книгу: " << books[num - 1] << "\n";
112      } else {
113          cout << "Цю книгу ніхто не позичав, щоб повертати.\n";
114      }
115
116      menu:
117
118      return;
119  }
120

```

```
Меню:
1 Позичити книгу
2 Повернути книгу
3 Перерахувати всі книги
4 Вийти
Я хочу: 1
Список книг:
1 Book1   В наявності
2 Book2   В наявності
3 Book3   В наявності
4 Book4   В наявності

Введіть номер книги, яку бажаєте взяти: 2
Ви взяли книгу: Book2
```

Хочете виконати іншу операцію? (y/n): y

```
Меню:
1 Позичити книгу
2 Повернути книгу
3 Перерахувати всі книги
4 Вийти
Я хочу: 3
Список книг:
1 Book1   В наявності
2 Book2   Не в наявності
3 Book3   В наявності
```

```
4 Book4   В наявності
```

Хочете виконати іншу операцію? (y/n): y

```
Меню:
1 Позичити книгу
2 Повернути книгу
3 Перерахувати всі книги
4 Вийти
Я хочу: 2
Список книг:
1 Book1   В наявності
2 Book2   Не в наявності
3 Book3   В наявності
4 Book4   В наявності
```

Введіть номер книги, яку бажаєте повернути: 3  
Цю книгу ніхто не позичав, щоб повертати.

Хочете виконати іншу операцію? (y/n): y

```
Меню:
1 Позичити книгу
2 Повернути книгу
3 Перерахувати всі книги
4 Вийти
Я хочу: 4
```

Хочете виконати іншу операцію? (y/n): n



## Practice# programming: Self Practice Task

Вам задається пісенька, яку мутикав собі під ніс Колобок. Необхідно знайти кількість різних слів, що зустрічаються в ній.

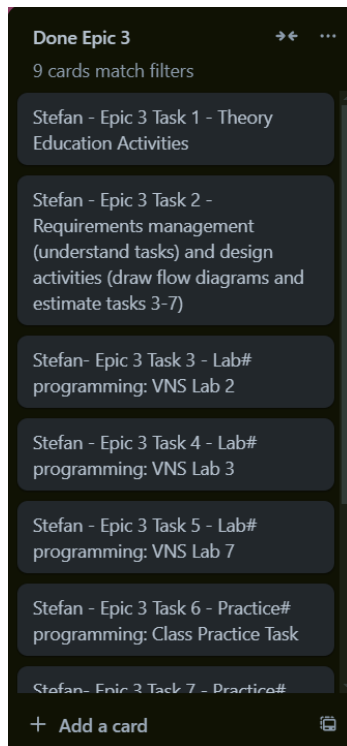
Слова в пісні розділені пробілами. Два слова вважаються однаковими, якщо вони складаються з однакової кількості букв та відповідні букви є однаковими. Велика та маленька відповідні букви вважаються однаковими.

```
1 //algotester 0631 kolobok :)
2
3 #include <iostream>
4 #include <string>
5 #include <cstring>
6 #include <sstream>
7 #include <cctype>
8 #include <set>
9
10 using namespace std;
11
12 int main(){
13     string song;
14     getline (cin, song);
15
16     for(char &c : song){
17         c = tolower(c);
18     }
19
20     set<string> set;
21     string word;
22     stringstream strStream(song);
23
24     while(strStream >> word){
25         set.insert(word);
26     }
27
28     int unique = set.size();
29
30     cout << unique;
31 }
```

```
graph TD
    Start([start]) --> Input[/input a song/]
    Input --> CharCheck{Any chars left in a song?}
    CharCheck -- yes --> ToLower[tolower(char)]
    ToLower --> CharCheck
    CharCheck -- no --> CreateSet[create a set for storing unique words, string stream and a tmp word var]
    CreateSet --> WordCheck{any words left in a stream?}
    WordCheck -- yes --> GetWord[get the next word from stream and insert it into the set]
    GetWord --> WordCheck
    WordCheck -- no --> CalcSize[calculate the size of the set of unique words]
    CalcSize --> Print[/print the number of unique words/]
    Print --> End([end])
```

| Created      | Problem        | Compiler | Result   | Time (sec.) | Memory (MiB) | #       |
|--------------|----------------|----------|----------|-------------|--------------|---------|
| 16 hours ago | 0631 - Колобок | C++ 23   | Accepted | 0.003       | 1.313        | 1845139 |

## Trello



## [Pull request](#)

**Висновок:** За лабораторну роботу я навчився використовувати різні типи циклів і управляти їх виконанням, створювати функції з різними параметрами, застосовувати рекурсію, а також зрозумів принципи використання просторів імен і перевантаження функцій у програмуванні.