Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



# Звіт

про виконання

**Лабораторних та практичних робіт №3*:*** «Мови та парадигми програмування»

***з розділу***: «Лінійні та розгалужені алгоритми. Умовні оператори. Константи,

змінні»

***Виконав:***

студент групи ШІ-11 Сабадило Марко

Львів 2023

**Тема роботи:**  Ітерація

**Мета роботи:**

**Теоретичні відомості:**

Цикли в C++ - це конструкції, які дозволяють повторювати блок коду декілька разів. Цикл for визначає ініціалізацію, умову продовження та крок ітерації. Цикл while повторюється, доки задана умова є істинною. Цикл do-while виконується принаймні один раз, а потім повторюється, доки умова залишається істинною.

**Джерела:**

**- cppreference.com - C++ for Loop**

**- cppreference.com - C++ while Loop**

**- cppreference.com - C++ do-while Loop**

**Виконання роботи:**

1. **Опрацювання завдання та вимог до програм та середовища:**

Завдання №1 VNS Lab 1 - Task 1-1

* + Варіант завдання - 2
  + Деталі завдання
    1. Необхідно обчислити результат дробу з двома невідомими, що вводяться користувачем
    2. Необхідно порівняти результати обчислень з використанням типів float і double
  + Важливі деталі для врахування в імплементації програми
    1. Важливо розбити приклад на вирази, і зберігати значення у змінних для уникнення помилок, зручності програміста Завдання №2 VNS Lab 1 - Task 1-2
  + Варіант завдання - 2
  + Деталі завдання
  1. Необхідно отримати результати дій кількох операндів на дві змінні, де може існувати двозначність
  2. Необхідно дослідити дані результати і пояснити пріоритетність операндів
  + Важливі деталі для врахування в імплементації програми

Існування директиви boolalpha для налаштування потоків виводу (не потрібно виводити стрічку “true” через if, можна виводити логічні змінні напряму) Завдання № 3 Algotester Lab 1

* + Варіант завдання - 2
  + Деталі завдання

Дано чотири числа, які відповідають довжині ніжок столу до відпилювання. Потрібно визначити, чи можна відпиляти від кожної ніжки довжину, яка відповідає даному числу, так, щоб стіл не перевернувся і залишився паралельним підлозі.

* + Важливі деталі для врахування в імплементації програми

Якщо довжина відпилювання перевищує довжину ноги, то вивести ERROR. Завдання № 4 Class practice work

* + Деталі завдання

1. **Дизайн та планована оцінка часу виконання завдань:**

Програма № 1

* + Блок-схемаЗображення, що містить текст, знімок екрана, схема, ряд

    Автоматично згенерований опис
  + Планований час на реалізацію - 15 хвилин

Зображення, що містить дизайн, лампа, мистецтво, чорно-білий

Автоматично згенерований опис

Зображення, що містить текст, знімок екрана, дизайн, Шрифт

Автоматично згенерований опис

Зображення, що містить знімок екрана, текст, чорно-білий, ряд

Автоматично згенерований опис

**3. Код програм з посиланням на зовнішні ресурси:**

Завдання № 1

|  |
| --- |
| #include <iostream> #include <cmath> using namespace std; // Функція для обчислення факторіалу double factorial(int n) {  if (n == 0 || n == 1) {  return 1;  } else {  return n \* factorial(n - 1);  } }  int main() {  int n = 13; // Кількість членів ряду  double sum = 0.0;   for (int i = 1; i <= n; ++i) {  double term = log(factorial(i)) / pow(i, 2);  sum += term;  }   cout << sum <<endl;   return 0; } |

[Посилання на Pull reqest](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/979/files#diff-5bc67772fc3c19f98dd6d241189e59f71880cb0eb7cff05e89abefd6849526a6)

Завдання № 2

|  |
| --- |
| #include <iostream> #include <cmath>  using namespace std;  double arctanTaylorSeries(double x, int n) {  double result = 0.0;  int sign = 1;   for (int i = 1; i <= n; i += 2) {  result += sign \* pow(x, i) / i;  sign \*= -1;  }   return result; }  double arctanWithAccuracy(double x, double epsilon) {  double result = 0.0;  double term = x;  int i = 1;   while (fabs(term) > epsilon) {  result += term;  term = -term \* x \* x / (2 \* i + 1);  ++i;  }   return result; }  int main() {  double a = 0.1;  double b = 1.0;  int k = 10;   for (double x = a; x <= b; x += (b - a) / k) {  // точне значення arctan(x)  double exactValue = atan(x);   // а) для заданого n  int n = 10; // можна змінювати  double resultWithN = arctanTaylorSeries(x, n);   // б) для заданої точності ε  double epsilon = 0.0001; // можна змінювати  double resultWithAccuracy = arctanWithAccuracy(x, epsilon);   cout << "x = " << x << "\n";  cout << "Exact value: " << exactValue << "\n";  cout << "Approximation with n=" << n << ": " << resultWithN << "\n";  cout << "Approximation with accuracy e=" << epsilon << ": " << resultWithAccuracy << "\n\n";  }   return 0; } |

[Посилання на Pull reqest](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/979/files#diff-b9e74c2901acd887edacebb9498766dd1a74e6d9de628e21b6cb03af67bdbce3)

Завдання № 3

|  |
| --- |
| # #include <iostream> #include <cmath> #include <cstdarg>  using namespace std;  // Функція для обчислення кутів трикутника за його сторонами double triangleAngle(double a, double b, double c) {  double angleRad = acos((a \* a + b \* b - c \* c) / (2 \* a \* b));  return angleRad \* 180 / M\_PI; // Перетворення радіан в градуси }  // Функція для обчислення кутів n-кутника за заданими сторонами double angles(int sides, ...) {  va\_list args;  va\_start(args, sides);   double sumOfAngles = 0.0;  for (int i = 0; i < sides - 2; ++i) {  double side = va\_arg(args, double);  double nextSide = va\_arg(args, double);  double thirdSide = va\_arg(args, double);   sumOfAngles += triangleAngle(side, nextSide, thirdSide);  }   va\_end(args);   return sumOfAngles; }  int main() {  // Знаходження кутів трикутника  double angleTriangle = angles(3, 3.0, 4.0, 5.0);  cout << angleTriangle << endl;   // Знаходження кутів 9-кутника  double anglePolygon9 = angles(9, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0);  cout << anglePolygon9 << endl;   // Знаходження кутів 11-кутника  double anglePolygon11 = angles(11, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0, 11.0);  cout << anglePolygon11 << endl;   return 0; } |
|  |

[Посилання на pullreqet](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/979/files#diff-c5ee8c9520d5bb020bf03b55285cb998ecc217d9e374652073dc49a20960ad82)

Завдання № 3.2

|  |
| --- |
| #include <iostream> #include <cstring> // Для роботи з рядками using namespace std;  // a) Функція для знаходження кількості від'ємних елементів у масиві цілих чисел int count\_min\_el(const int arr[], int size) {  int count = 0;  for (int i = 0; i < size; ++i) {  if (arr[i] < 0) {  ++count;  }  }  return count; }  // b) Функція для знаходження кількості слів, що починаються і закінчуються на ту ж букву int count\_f\_let(const char\* str) {  int count = 0;  char\* strCopy = new char[strlen(str) + 1]; // Створюємо копію рядка  strcpy(strCopy, str);   char\* token = strtok(strCopy, " ,.;");  while (token != nullptr) {  if (strlen(token) >= 2 && token[0] == token[strlen(token) - 1]) {  ++count;  }  token = strtok(nullptr, " ,.;");  }   delete[] strCopy; // Важливо видалити виділену пам'ять   return count; }  int main() {  // a) Приклад використання для масиву цілих чисел  int intArray[] = {1, -2, 3, -4, 5, -6};  int intArraySize = sizeof(intArray) / sizeof(int);   int negativeCount = count\_min\_el(intArray, intArraySize);  cout << negativeCount << endl;   // b) Приклад використання для рядка  const char\* sampleString = "apple banana level civic cat dog";  int wordCount = count\_f\_let(sampleString);  cout << wordCount <<endl;   return 0; } |
|  |

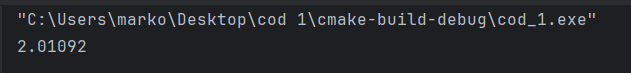
[Посидання на Pullreqest](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/979/files#diff-c5ee8c9520d5bb020bf03b55285cb998ecc217d9e374652073dc49a20960ad82)

Завдання № 4

|  |
| --- |
| # #include <iostream> #include <string> #include <vector> using namespace std; class Book { public:  string title;  string author;   // Конструктор для ініціалізації книги  Book(const string& t, const string& a) : title(t), author(a) {}   // Перевантажений оператор виведення для виводу книги  friend ostream& operator<<(ostream& os, const Book& book) {  os << "Title: " << book.title << "\nAuthor: " << book.author << "\n";  return os;  } };  // Функція для вибору книжки зі списку Book chooseBook(const vector<Book>& books, int index) {  if (index >= 0 && index < books.size()) {  return books[index];  } else {  // Повертаємо порожню книгу у випадку невірного індексу  return Book("Invalid Book", "Invalid Author");  } }  int main() {  // Створюємо список книг  vector<Book> bookList;  bookList.push\_back(Book("Book 1", "Author 1"));  bookList.push\_back(Book("Book 2", "Author 2"));  bookList.push\_back(Book("Book 3", "Author 3"));   // Виводимо список книг для вибору  cout << "Choose a book from the list:\n";  for (int i = 0; i < bookList.size(); ++i) {  cout << i + 1 << ". " << bookList[i].title << "\n";  }   // Введення вибору користувача  int userChoice;  cout << "Enter the number of the book you want to choose: ";  cin >> userChoice;   // Виводимо вибрану книгу  Book selectedBook = chooseBook(bookList, userChoice - 1);  cout << "\nYou have chosen:\n" << selectedBook;   return 0; } |

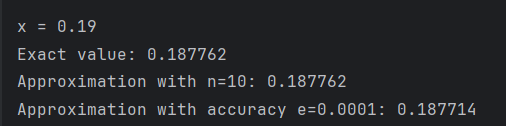
[Посилання на Pull reqest](https://github.com/artificial-intelligence-department/ai_programming_playground/pull/979/files" \l "diff-b0b967432b1ddb44612c07c895d575871dc3dec518b172c6490a54ce45c8f002)

Завдання № 1



Час, витрачений на виконання завдання - 20 хвилин

Завдання № 2



Час, витрачений на виконання завдання - 20 хвилин

Завдання № 3

Зображення, що містить текст, Шрифт, знімок екрана

Автоматично згенерований опис

Час, витрачений на виконання завдання - 20 хвилин

Завдання № 3.2

Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис

Час, витрачений на виконання завдання - 20 хвилин

Завдання № 4

Зображення, що містить текст, знімок екрана, Шрифт, алгебра

Автоматично згенерований опис