

Міністерство освіти і науки України

Державний заклад
«Південноукраїнський національний педагогічний університет
імені К. Д. Ушинського»

Фізико-математичний факультет

Кобякова Людмила Миколаївна
Шувалова Ольга Ігорівна

Методичні рекомендації
«Частина 1. С++: Керуючі конструкції»
для самостійної підготовки і виконання лабораторних робіт
навчального курсу ОК 10 «Інформатика»
для здобувачів першого (бакалаврського) рівня вищої освіти
1 року навчання спеціальності 014.09 Середня освіта (інформатика)

УДК: 004.42+004.432

Друкується за рішенням Вченої ради Державного закладу «Південноукраїнський національний педагогічний університет імені К.Д.Ушинського» (протокол № ____ від « » 2021 року).

Кобякова Л.М., Шувалова О.І.. **C++: Керуючі конструкції:** Методичні рекомендації для самостійної підготовки і виконання лабораторних робіт навчального курсу ОК 10 «Інформатика» для здобувачів першого (бакалаврського) рівня вищої освіти 1-го року навчання спеціальності 014.09 Середня освіта (інформатика). Одеса: ПНПУ імені К.Д.Ушинського, 2021. 52 с.

Рецензенти:

Мазурок Тетяна Леонідівна, доктор технічних наук, професор, завідувач кафедри прикладної математики та інформатики Південноукраїнського національного педагогічного університету імені К. Д. Ушинського.

Сметаніна Людмила Сергіївна, кандидат педагогічних наук, доцент, доцент кафедри електронного урядування та інформаційних систем Одеського регіонального інституту державного управління

Поданий у методичних рекомендаціях матеріал охоплює питання тем «Лінійний алгоритм», «Алгоритм розгалуження», «Циклічний алгоритм» мовою C++, що формують уявлення про методику розробки програм, є фундаментом програмування і важливою ланкою професійної освіти для спеціалістів у галузі прикладної математики та інформатики.

Реалізація навчальних прикладів акцентує увагу на принципах побудови ефективних алгоритмів, способах опису вхідних і вихідних даних для вирішення прикладних задач.

ЗМІСТ

НАВЧАЛЬНІ МАТЕРІАЛИ	5
Творець C++ Б'ярн Страуструп	5
Історія створення і розвитку мови програмування C++	5
Сайти з навчальними матеріалами. Уроки	5
Відеоматеріали. Відеоуроки	5
Самовчителі. Підручники	5
Довідники	5
Задачники, практикуми	5
Статті	6
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ	8
ВСТУП	9
ТЕМА 1: ЛІНІЙНИЙ АЛГОРИТМ	10
Лабораторна робота №1	10
1.1. Локалізація та кирилиця в консолі	10
1.2. Структура програми	10
1.3. Типи даних. Змінні. Константи	10
1.4. Статична типізація та перетворення типів	10
1.5. Бібліотека <code>cmath</code> (математичні константи та функції). Округлен-	
ня. Арифметичні операції та вирази	10
1.6. Введення з клавіатури. Вивід на екран	10
1.7. Простори найменувань та <code>using</code>	10
1.8. Операції присвоювання	10
1.9. Розв'язання задач	11
1.10. Стандартний алгоритм обміну значеннями змінних	14
1.11. Задачі на написання програмного коду	15
1.12. Задачі на читання програмного коду	16
1.13. Відповіді до задач на читання програмного коду	20
1.14. Домашнє завдання	21
ТЕМА 2. АЛГОРИТМ РОЗГАЛУЖЕННЯ	22
Лабораторна робота №2	22
2.1. Умовні вирази. Операції порівняння. Логічні операції	22
2.2. Умовні конструкції: умовний оператор, оператор <code>switch</code> , тернар-	22
ний оператор	
2.3. Розв'язання задач	22
2.4. Задачі на написання програмного коду	27
2.5. Задачі на читання програмного коду	28
2.6. Відповіді до задач на читання програмного коду	31
2.7. Домашнє завдання	33

ТЕМА 3. ЦИКЛІЧНИЙ АЛГОРИТМ	33
Лабораторна робота №3	33
3.1. Оператор циклу <code>for</code> . Оператори <code>continue</code> , <code>break</code>	33
3.1.1. Розв'язання задач	33
3.1.2. Задачі на написання програмного коду	38
3.1.3. Задачі на читання програмного коду	39
3.1.4. Відповіді до задач на читання програмного коду	42
3.1.5. Домашнє завдання	43
Лабораторна робота №4	43
3.2. Оператори циклу <code>while</code> , <code>do-while</code>	43
3.2.1. Розв'язання задач	44
3.2.2. Стандартний алгоритм Евкліда	45
3.2.3. Задачі на написання програмного коду	46
3.2.4. Задачі на читання програмного коду	47
3.2.5. Відповіді до задач на читання програмного коду	48
3.2.6. Домашнє завдання	48
Лабораторна робота №5	49
3.3. Вкладені цикли	49
3.3.1. Розв'язання задач	49
3.3.2. Задачі на написання програмного коду	51
3.3.3. Домашнє завдання	52

НАВЧАЛЬНІ МАТЕРІАЛИ

Творець C++ Б'ярн Страуструп

https://uk.wikipedia.org/wiki/Б%27ярн_Страуструп

Історія створення і розвитку мови програмування

<https://uk.wikipedia.org/wiki/С%2В%2В>

Сайти з навчальними матеріалами. Уроки

<https://metanit.com/cpp/tutorial/>

<https://code-live.ru/tag/cpp-manual/>

<http://cppstudio.com/uk/>

<https://ravesli.com/uroki-cpp/>

<https://acode.com.ua/uroki-po-cpp/>

Відеоматеріали. Відеоуроки

Канал [#SimpleCode](#) на YouTube

<https://www.youtube.com/playlist?list=PLQOaTSbfxUtCrKs0nicOg2npJQYSPG09r>

Самовчителі. Підручники:

1. Страуструп Бьярне. Программирование: принципы и практика с использованием C++. М., ООО "И.Д. Вильямс", 2016. 1328 с.

2. Липман Стенли, Лажойе Жози, Му Барбара Э. Язык программирования C++: Базовый курс. М., ООО "И.Д. Вильямс", 2014. 1120 с.

Довідники

<https://docs.microsoft.com/ru-ru/cpp/cpp/cpp-language-reference?view=msvc-160>

<https://en.cppreference.com/w/>

Задачники, практикуми

1. Абрамов С.А и др. Задачи по программированию. Вологда, 2000. 596с.
<https://ideafix.name/wp-content/uploads/stuff/book95.pdf>

2. Абрамян М.Э. 1000 задач по программированию. Ростов-на-Дону, 2004.

<http://www.ptaskbook.com/ru/>

3. Електронний підручник Рішельєвського ліцею (ЕПРЛ)
<http://rlwestudy.blogspot.com/p/powerpoint.html>

4. Златопольский Д. 1400 задач по программированию. М., ДМК Пресс, 2020. 192 с.

<https://coderoll.net/dev-book/200-1400-zadach-po-programirovaniyu.html>

Статті

1) Компілятор

<https://uk.wikipedia.org/wiki/Компілятор>

2) Алгоритм

<https://ru.wikipedia.org/wiki/Алгоритм>

3) Поняття алгоритму

https://ru.wikibooks.org/wiki/Что_такое_алгоритм

4) Блок-схема

<https://uk.wikipedia.org/wiki/Блок-схема>

5) Типи даних в C++ / Built-in types (C++)

<https://docs.microsoft.com/en-us/cpp/cpp/fundamental-types-cpp?view=msvc-160&viewFallbackFrom=vs-2019>

6) Оператори в мові C++ / C++ built-in operators, precedence, and associativity

<https://docs.microsoft.com/en-us/cpp/cpp/cpp-built-in-operators-precedence-and-associativity?view=msvc-160&viewFallbackFrom=vs-2019>

7) Перетворення типів в C / Type Conversions (C)

<https://docs.microsoft.com/en-us/cpp/c-language/type-conversions-c?redirectedfrom=MSDN&view=msvc-160>

8) Явне перетворення типів / Type-Cast Conversions

<https://docs.microsoft.com/en-us/cpp/c-language/type-cast-conversions?redirectedfrom=MSDN&view=msvc-160>

9) l-value та r-value / Lvalues and Rvalues (C++)

<https://docs.microsoft.com/en-us/cpp/cpp/lvalues-and-rvalues-visual-cpp?view=msvc-160&viewFallbackFrom=vs-2019>

10) if else / if-else statement (C++)

<https://docs.microsoft.com/en-us/cpp/cpp/if-else-statement-cpp?view=msvc-160&viewFallbackFrom=vs-2019>

11) Конструкція switch / switch statement (C++)

<https://docs.microsoft.com/en-us/cpp/cpp/switch-statement-cpp?view=msvc-160&viewFallbackFrom=vs-2019>

12) Цикл while / while Statement (C++)

<https://docs.microsoft.com/en-us/cpp/cpp/while-statement-cpp?view=msvc-160&viewFallbackFrom=vs-2019>

13) Цикл do while / do-while Statement (C++)

<https://docs.microsoft.com/en-us/cpp/cpp/do-while-statement-cpp?view=msvc-160&viewFallbackFrom=vs-2019>

14) Цикл for / for statement (C++)

<https://docs.microsoft.com/en-us/cpp/cpp/for-statement-cpp?view=msvc-160>

15) Використання break / break Statement (C++)

<https://docs.microsoft.com/en-us/cpp/cpp/break-statement-cpp?view=msvc-160&viewFallbackFrom=vs-2019>

16) Використання continue / continue Statement (C++)

<https://docs.microsoft.com/en-us/cpp/cpp/continue-statement-cpp?view=msvc-160&viewFallbackFrom=vs-2019>

17) Нескінченний цикл

https://ru.wikipedia.org/wiki/Бесконечный_цикл

18) Що таке goto, використовувати його чи ні?

<https://ru.wikipedia.org/wiki/Goto>

19) Спагетті-код

<https://ru.wikipedia.org/wiki/Спагетти-код>

20) Спагетті-стаття про спагетті-код

<https://habr.com/ru/post/187154/>

21) Отладчик та принципи його використання / Navigating through Code with the Debugger

<https://docs.microsoft.com/ru-ru/previous-versions/visualstudio/visual-studio-2015/debugger/navigating-through-code-with-the-debugger?view=vs-2015&redirectedfrom=MSDN>

22) Прийоми для роботи з отладчиком / 5 Awesome Visual Studio Debugger Features

https://www.cprogramming.com/tutorial/visual_studio_tips.html

23) Цикл статей про використання отладчика / Debugging with Visual Studio 2005/2008, Part 1: Debugging Concepts

https://www.cprogramming.com/tutorial/debugging_concepts.html

24) Путівник по отладчику / Debugger Basics

<https://docs.microsoft.com/ru-ru/previous-versions/visualstudio/visual-studio-2015/debugger/debugger-basics?view=vs-2015&redirectedfrom=MSDN>

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

1. онлайн-компілятори

1) **ideone**

ideone.com

обрати C++ 14

2) **replit**

<https://replit.com/languages/cpp>

3) **C++ shell**

<http://cpp.sh>

2. MS Visual Studio

Інструкція із встановлення

<https://metanit.com/cpp/tutorial/1.4.php>

3. Qt Creator

Інструкція із встановлення

<http://cppstudio.com/post/9273/>

ВСТУП

Навчальна дисципліна ОК 10 «Інформатика» для здобувачів першого (бакалаврського) рівня вищої освіти 1-го року навчання спеціальності 014.09 Середня освіта (інформатика) за навчальним планом підготовки фахівця охоплює 12 кредитів загальною кількістю 360 годин. Аудиторні заняття – 180 годин (лекції – 36 годин, лабораторні роботи — 144 години), 180 годин відведено на самостійну роботу студентів. Зміст навчальної дисципліни розділено за програмою на 2 модуля, перший присвячено основам інформатики, другий охоплює питання процедурного програмування мовою програмування C++ та/або Python на вибір студента. Вибір студентом мови програмування дозволяє реалізувати індивідуальний підхід до навчання програмуванню.

Методичні рекомендації «C++: Керуючі конструкції:» розкривають питання тем «Лінійний алгоритм», «Алгоритм з розгалуженням», «Циклічний алгоритм» навчальної програми другого модуля навчальної дисципліни ОК 10 «Інформатика». За програмою навчальної дисципліни заняття розраховані на 30 годин аудиторної і 30 годин самостійної роботи.

Зміст не потребує додаткових базових знань, частково спирається на шкільний курс математики.

Розроблені лабораторні роботи є основою для засвоєння процедурного програмування, а в подальшому – об'єктно-орієнтованого програмування та практичним підґрунтям для дисциплін комп'ютерного циклу.

Тема 1. ЛІНІЙНИЙ АЛГОРИТМ

ЛАБОРАТОРНА РОБОТА №1

- 1.1. Локалізація та кирилиця в консолі
- 1.2. Структура програми
- 1.3. Типи даних. Змінні. Константи
- 1.4. Статична типізація та перетворення типів
- 1.5. Бібліотека `cmath` (математичні константи та функції). Округлення. Арифметичні операції та вирази
- 1.6. Введення з клавіатури. Вивід на екран
- 1.7. Простори найменувань та `using`
- 1.8. Операції присвоювання
- 1.9. Розв'язання задач
- 1.10. Стандартний алгоритм обміну значеннями змінних
- 1.11. Задачі на написання програмного коду
- 1.12. Задачі на читання програмного коду
- 1.13. Відповіді до задач на читання програмного коду
- 1.14. Домашнє завдання

ТЕОРЕТИЧНИЙ МАТЕРІАЛ:

- 1.1. Локалізація та кирилиця в консолі

<https://metanit.com/cpp/tutorial/1.5.php>

- 1.2. Структура програми

<https://metanit.com/cpp/tutorial/2.1.php>

- 1.3. Типи даних. Змінні. Константи

<https://metanit.com/cpp/tutorial/2.2.php>

<https://metanit.com/cpp/tutorial/2.3.php>

<https://metanit.com/cpp/tutorial/2.5.php>

- 1.4. Статична типізація та перетворення типів

<https://metanit.com/cpp/tutorial/2.4.php>

- 1.5. Бібліотека `cmath` (математичні константи і функції). Округлення.

Арифметичні операції та вирази

<http://cppstudio.com/post/413/>

<https://proginform.ru/round/>

<https://metanit.com/cpp/tutorial/2.6.php>

- 1.6. Введення з клавіатури. Вивід на екран

<https://metanit.com/cpp/tutorial/2.10.php>

- 1.7. Простори найменувань та `using`

<https://metanit.com/cpp/tutorial/2.10.php>

- 1.8. Операції присвоювання

<https://metanit.com/cpp/tutorial/2.9.php>

1.9. Розв'язання задач

1. Дано 2 цілих числа. Знайти їх суму, різницю, добуток, результат цілочисельного ділення, залишок від ділення, частку від звичайного ділення першого числа на друге.

Алгоритм:

- 1) оголошення та завдання значень змінних a і b ;
- 2) оголошення змінної sum , обчислення суми чисел a і b , вивід на екран;
- 3) оголошення змінної $difference$, обчислення різниці чисел a і b , вивід на екран;
- 4) оголошення змінної $product$, обчислення добутку чисел a і b , вивід на екран;
- 5) оголошення змінної int_div , обчислення частки цілочисельного ділення чисел a і b , вивід на екран;
- 6) оголошення змінної rem_div , обчислення залишку від цілочисельного ділення чисел a і b , вивід на екран;
- 7) оголошення змінної $division$, обчислення частки чисел a і b , вивід на екран.

Результат звичайного ділення 2-х чисел – дійсне число, тому змінна $division$ віднесена до типу $double$.

Для здійснення звичайного ділення хоча б один з операндів повинен бути дійсним числом, тому значення цілочисельної змінної a перетворено до дійсного типу: $(double)$ a .

	Програмний код
1	<code>#include <iostream></code>
2	<code>using namespace std;</code>
3	
4	<code>int main()</code>
5	<code>{</code>
6	<code>// 1) Оголошення, завдання значень змінним a і b,</code>
7	<code>// вивід на екран</code>
8	<code>int a = 25, b = 10;</code>
9	<code>cout << "a = " << a << " b = " << b << endl;</code>
10	
11	<code>// 2) Оголошення змінної sum, обчислення суми чисел a і b,</code>
12	<code>// вивід на екран</code>
13	<code>int sum = a + b;</code>
14	<code>cout << "sum = " << sum << endl;</code>
15	
16	<code>// 3) Оголошення змінної $difference$, обчислення різниці чисел</code>
17	<code>// a і b, вивід на екран</code>
18	<code>int difference = a - b;</code>
19	<code>cout << "difference = " << difference << endl;</code>
20	

```

21 // 4) Оголошення змінної product, обчислення добутку чисел a і
22 //    b, вивід на екран
23     int product = a * b;
24     cout << "product = " << product << endl;
25
26 // 5) Оголошення змінної int_div, обчислення частки від
27 //    цілочисельного ділення чисел a і b, вивід на екран
28     int int_div = a / b;
29     cout << "int_div = " << int_div << endl;
30
31 // 6) Оголошення змінної rem_div, обчислення залишку від
32 //    цілочисельного ділення чисел a і b, вивід на екран
33     int rem_div = a % b;
34     cout << "rem_div = " << rem_div << endl;
35
36 // 7) Оголошення змінної division, обчислення частки від
37 //    звичайного ділення чисел a і b, вивід на екран
38     double division = (double) a / b;
39     cout << "division = " << division << endl;
40
41     return 0;
42 }

```

На екрані:

```

a = 25  b = 10
sum = 35
difference = 15
product = 250
int_div = 2
rem_div = 5
division = 2.5

```

2. [Абрамян, Boolean34-36]. Дано координати 2-х різних полів шахівниці x_1, y_1, x_2, y_2 (цілі числа, які належать діапазону 1–8). Враховуючи, що ліве нижнє поле шахівниці (1, 1) чорне, перевірити істинність висловлювань:

поле (x_1, y_1) біле;

поле (x_2, y_2) чорне;

поля однакового кольору.

тура за один хід може перейти з першого поля на друге.

Алгоритм:

- 1) оголошення цілих змінних x_1, y_1, x_2, y_2
- 2) вивід на екран повідомлення «Введіть x та y координати 1-го поля шахівниці» та введення з клавіатури значень змінних x_1, y_1 ;
- 3) вивід на екран повідомлень «Введіть x та y координати 2-го поля шахівниці» та введення з клавіатури значень змінних x_2, y_2 ;
- 4) оголошення та обчислення значень булевих змінних `white1` та `black2`

Клітка біла, якщо x-ва та y-ва координати клітки різної парності, чорна,

якщо однакова.

Якщо число a парне, залишок від ділення на 2 дорівнює 0 ($a \% 2 == 0$), якщо непарне – 1 ($a \% 2 == 1$),

5) оголошення та обчислення булевої змінної `colors_equals`

Якщо змінні `white1` та `black2` мають різні логічні значення (одне з них `true`, друге – `false`);

6) оголошення та обчислення булевої змінної `ladjya_move`

Тура ходить вертикально, тоді x -ві координати кліток співпадають ($x1 == x2$), або горизонтально, тоді співпадають y -ві ($y1 == y2$).

	Програмний код
1	<code>#include <iostream></code>
2	<code>using namespace std;</code>
3	
4	<code>int main()</code>
5	<code>{</code>
6	<code>// 1) Оголошення цілих змінних x1, y1, x2, y2</code>
7	<code>int x1, y1, x2, y2;</code>
8	
9	<code>// 2)-5) Введення з клавіатури значень змінних x1, y1, x2, y2</code>
10	<code>cout << "Введіть x та y координати 1-го поля шахівниці"</code>
11	<code><< endl;</code>
12	<code>cin >> x1 >> y1;</code>
13	<code>cout << "Введіть x та y координати 2-го поля шахівниці"</code>
14	<code><< endl;</code>
15	<code>cin >> x2 >> y2;</code>
16	
17	<code>// 6) З'ясування кольорів кліток</code>
18	<code>cout << "0 - false, 1 - true" << endl;</code>
19	
20	<code>bool white1 = (x1 % 2 == 1 && y1 % 2 == 0 x1 % 2 == 0</code>
21	<code>&& y1 % 2 == 1);</code>
22	<code>cout << "1-я клітка біла? " << white1 << endl;</code>
23	<code>bool black2 = (x2 % 2 == 0 && y2 % 2 == 0 x2 % 2 == 1</code>
24	<code>&& y2 % 2 == 1);</code>
25	<code>cout << "2-я клітка чорна? " << black2 << endl;</code>
26	
27	<code>// 7) Чи співпадають кольори кліток?</code>
28	<code>bool colors_equals = white1 != black2;</code>
29	<code>cout << "Кольори 2-х кліток співпадають? " <<</code>
30	<code>colors_equals << endl;</code>
31	
32	<code>// 8) Чи може тура за один хід перейти з 1-ої клітки на 2-у?</code>
33	<code>bool ladjya_move = (x1 == x2) (y1 == y2);</code>
34	<code>cout << "Чи може тура за один хід дійти з 1-ої клітки у</code>
35	<code>2-у? " << ladjya_move << endl;</code>
36	
37	<code>return 0;</code>
38	<code>}</code>

На екрані:

```

Введіть x та y координати 1-го поля шахівниці
1 2
Введіть x та y координати 2-го поля шахівниці
1 8
0 - false, 1 - true
1-я клітка біла?      1
2-я клітка чорна?    0
Кольори 2-х кліток співпадають?    1
Чи може тура за один хід дійти з 1-ої клітки у 2-у?    1

```

3. Дано 2 сторони трикутника та кут (у градусах) між ними. Знайти протилежну сторону.

Алгоритм:

- 1) підключити бібліотеку математичних функцій `<cmath>`
- 2) ініціалізувати вихідні дані – сторони a , b та γ – кут між ними (у градусах)
- 3) перевести кут з градусної міри в радіанну

$$\gamma' = \gamma \cdot \frac{\pi}{180^\circ}, \quad \gamma' - \text{кут у радіанах, } \gamma - \text{кут в градусах}$$
 M_PI – стандартна константа π з бібліотеки `<cmath>`
- 4) обчислити за теоремою косинусів 3-ю. сторону трикутника c та вивести її на екран

$$c = \sqrt{a^2 + b^2 - 2ab \cos \gamma}$$

	Програмний код
1	<code>#include <iostream></code>
2	<code>#include <cmath> //підключення бібліотеки математичних функцій</code>
3	<code>using namespace std;</code>
4	
5	<code>int main()</code>
6	<code>{</code>
7	<code>// 1) ініціалізація вихідних даних</code>
8	<code>double a = 10, b = 7; // сторони трикутника</code>
9	<code>int angle_gradus = 60; // кут у градусах</code>
10	
11	<code>// 2) обчислення радіанної міри кута</code>
12	<code>double angle_radian = angle_gradus * M_PI / 180;</code>
13	
14	<code>// 3) обчислення 3-ї сторони трикутника та вивід на екран</code>
15	<code>double c = sqrt(a*a + b*b - 2*a*b*cos(angle_radian));</code>
16	<code>cout << "c = " << c << endl;</code>
17	
18	<code>return 0;</code>
19	<code>}</code>

На екрані:

```
c = 8.88819
```

1.10. Стандартний алгоритм обміну значеннями змінних

Дано 2 змінних $a = 100$ и $b = 1$.

Здійснити обмін значеннями змінних, тобто. в результаті: $a = 1$, $b = 100$.

Невірно		Вірно	
Програмний код	Результат виконання програмного коду	Програмний код	Результат виконання програмного коду
$b = a;$ $a = b;$	$b = 100$ $a = 100$	$temp = a;$ $a = b;$ $b = temp;$	$temp = 100$ $a = 1$ $b = 100$

$temp$ – допоміжна змінна (3-я каструлька при обміні вмісту 2 каструлек ☺).

1.11. Задачі на написання програмного коду

1. [Абрамян, Begin8-9]. Дано 3 цілих числа a , b , c . Знайти їх середнє арифметичне та середнє геометричне (квадратний корінь їх добутку).

2. [Абрамян, Begin11-12]. Дано 3 дійсних числа. Знайти

– різницю модулів та добуток їх квадратів;

– модуль їх суми та квадрат їх частки.

3. [Абрамян, Begin24]. Дано змінні a , b , c . Змінити їх значення, перемістивши вміст a в c , c — в b , b — в a , та вивести вихідні і нові значення змінних a , b , c на екран.

4. [Абрамян, Begin28]. Дано число a . Обчислити a^{15} , використовуючи дві допоміжні змінні та 5 операцій множення. Для цього послідовно знаходити a^2 , a^3 , a^5 , a^{10} , a^{15} . Вивести всі знайдені ступені числа a .

5. [Абрамян, Integer3]. Дано розмір файлу в байтах. Використовуючи операцію ділення націло, знайти кількість повних кілобайтів, які займає даний файл (1 кілобайт = 1024 байта).

6. [Абрамян, Integer11]. Дано тризначне число. Знайти суму та додатак його цифр.

7. [Абрамян, Integer19-23]. З начала доби пройшло N секунд (N — ціле). Знайти

– кількість повних хвилин, які минули від начала доби;

– кількість повних годин, які минули від начала доби;

– кількість секунд, які минули від начала останньої хвилини;

– кількість секунд, які минули від начала останньої години;

– кількість повних хвилин, які минули від начала останньої години.

8. [Абрамян, Integer 24]. Дні тижня пронумеровані наступним чином: 0 — неділя, 1 — понеділок, 2 — вівторок, ..., 6 — субота. Дано ціле число K , яке належить діапазону 1–365. Визначити номер дня тижня для K -го дня року, якщо відомо, що в цьому році 1 січня було днем з номером N , N введіть з клавіатури.

9. Дано дійсні та уявні частини 2-х комплексних чисел. Знайти дійсні та

уявні частини їх суми, добутку та частки.

10. Дано гіпотенузу та гострий кут у градусах прямокутного трикутника. Знайти його катети.

11. Дано катет та його проекція на гіпотенузу. Знайти гіпотенузу, висоту, другий катет та його проекцію на гіпотенузу.

12. Дано перший член a_1 , різниця d й кількість n перших членів арифметичної прогресії. Обчисліть n -ий член a_n та суму n перших членів арифметичної прогресії s_n .

13. [Шень, 1.1.9]. Послідовність Фібоначчі визначається формулами: $a_0 = 0$, $a_1 = 1$, $a_k = a_{k-1} + a_{k-2}$ при $k \geq 2$. Обчисліть a_2 , a_3 , a_4 .

1.12. Задачі на читання програмного коду

1. <pre>#include <iostream> using namespace std; int main() { double a = 10.1, b = 5.3; double p = 2 * (a + b); cout << "p = " << p << endl; double s = a * b; cout << "s = " << s << endl; return 0; }</pre>	2. <pre>#include <iostream> #include <cmath> using namespace std; int main() { double d = 10.1; double l = M_PI * d; cout << "l = " << l << endl; double r = d / 2; double s = M_PI*pow(r,2); cout << "s = " << s << endl; return 0; }</pre>
3. <pre>#include <iostream> #include <cmath> using namespace std; int main() { double a = 3, b = 4; double c = sqrt(a*a+b*b); cout << "c = " << c << endl; double p = a + b + c; cout << "p = " << p << endl; double pp = p / 2; double s = sqrt(pp*(pp-a) *(pp-b)*(pp-c)); cout << "s = " << s << endl; return 0; }</pre>	4. <pre>#include <iostream> #include <cmath> using namespace std; int main() { double x1 = 3, y1 = 0; double x2 = 0, y2 = 4; double x3 = 0, y3 = 0; double a = sqrt(pow(x2- x1, 2) + pow(y2-y1,2)); cout << "a = " << a << endl; double b = sqrt(pow(x3- x1, 2) + pow(y3-y1,2)); cout << "b = " << b << endl; double c = sqrt(pow(x3- x2, 2) + pow(y3-y2,2)); cout << "c = " << c << endl; }</pre>

			endl; return 0; }
5.	<pre>#include <cstdio> using namespace std; int main() { int a = 1, b = 100; printf("a = %d, b = %d\n", a, b); int temp = a; a = b; b = temp; printf("a = %d, b = %d\n", a, b); return 0; }</pre>	6.	<pre>#include <cstdio> using namespace std; int main() { int a=1, b = 10, c = 100; printf("a = %4d, b = %4d, c = %4d\n", a, b, c); int temp = c; c = b; b = a; a = temp; printf("a = %4d, b = %4d, c = %4d\n", a, b, c); return 0; }</pre>
7.	<pre>#include <cstdio> using namespace std; int main() { int a = 2; int p = a*a; p *= p; p *= p; printf("p = %d", p); return 0; }</pre>	8.	<pre>#include <iostream> using namespace std; int main() { int a = 42; int d1 = a / 10; int d2 = a % 10; int b = d2 * 10 + d1; cout << "a = " << a << " b = " << b << endl; return 0; }</pre>
9.	<pre>#include <iostream> using namespace std; int main() { double x1 = 1, y1 = 1; double x3 = 5, y3 = 4; double x2 = x1; double y2 = y3; double x4 = x3; double y4 = y1; cout << "(" << x1 << ", " << y1 << ")" << " -- "; cout << "(" << x2 << ", " << y2 << ")" << " -- "; cout << "(" << x3 << ", " << y3 << ")" << " -- "; cout << "(" << x4 << ", " << y4 << ")" << endl; }</pre>	10.	<pre>#include <iostream> #include <cmath> using namespace std; int main() { double aR = 2*M_PI/3; double c2aR = sqrt((1 + cos(aR))/2); cout << "cos(" << aR << " = " << c2aR << endl; double s2aR = sqrt((1 - cos(aR))/2); cout << "sin(" << aR << ") = " << s2aR << endl; return 0; }</pre>

	<pre> double a = abs(x3 - x1); double b = abs(y3 - y1); cout << "a = " << a << ", b = " << b << endl ; return 0; } </pre>		
11.	<pre> #include <iostream> #include <cmath> using namespace std; int main() { double a = 3, b = 4, c = 5; double aR = acos((c*c - a*a - b*b) / (2*a*b)); double aG = aR * 180 / M_PI; cout << "aG = " << aG << endl; return 0; } </pre>	12.	<pre> #include <iostream> using namespace std; int main() { double v = 10; double v0 = 3; int t1 = 5; int t2 = 3; double s = (v + v0) * t1 + (v - v0) * t2; cout << "s = " << s << endl; return 0; } </pre>
13.	<pre> #include <iostream> #include <cmath> using namespace std; int main() { double b1, q; int n; cout << "Введите b1: "; cin >> b1; cout << "Введите q: "; cin >> q; cout << "Введите n: "; cin >> n; double bn = b1 * pow(q, n-1); cout << "b[" << n << "] = " << bn << endl; double sn = b1*(1 - pow(q, n))/(1 - q); cout << "s[" << n << "] = " << sn << endl; return 0; } </pre>	14.	<pre> #include <iostream> using namespace std; int main() { int ch1 = 2, zn1 = 3; int ch2 = 3, zn2 = 4; int ch3 = ch1 * zn2 + ch2 * zn1; int zn3 = zn1 * zn2; cout << ch1 << "/" << zn1 << " + " << ch2 << "/" << zn2 << " = " << ch3 << "/" << zn3 << endl; int ch4 = ch1 * ch2; int zn4 = zn1 * zn2; cout << ch1 << "/" << zn1 << " * " << ch2 << "/" << zn2 << " = " << ch4 << "/" << zn4 << endl; int ch5 = ch1 * zn2; int zn5 = zn1 * ch2; cout << ch1 << "/" << zn1 << " : " << ch2 << "/" << zn2 << " = " << ch5 << "/" << zn5 << endl; return 0; } </pre>

15.	<pre> #include <iostream> #include <cmath> using namespace std; int main() { double p22 = sqrt(2); cout << "sqrt(2) = " << p22 << endl; double p33 = pow(3, 1./3); out << "pow(3, 1./3) = " << p33 << endl; double p44 = pow(4, (double) 1/4); cout << "pow(4, (double) 1/4) = " << p44 << endl; return 0; } </pre>	16. <pre> #include <iostream> #include <cmath> using namespace std; int main() { double a = 10.56; cout << "a = " << a << endl; cout << "floor(" << a << ") = " << floor(a) << endl; cout << "ceil(" << a << ") = " << ceil(a) << endl; cout << "round(" << a << ") = " << round(a) << endl; cout << "trunc(" << a << ") = " << trunc(a) << "\n\n"; double b = -a; cout << "b = " << b << endl; cout << "floor(" << b << ") = " << floor(b) << endl; cout << "ceil(" << b << ") = " << ceil(b) << endl; cout << "round(" << b << ") = " << round(b) << endl; cout << "trunc(" << b << ") = " << trunc(b) << endl; return 0; } </pre>
17.	<pre> #include <iostream> #include <cmath> using namespace std; int main() { double a = -1, b = 1; double r = sqrt(a*a + b*b); double ar = atan(b/a); cout << a << " + " << b << " i " << " = " << r << " (cos(" << ar << ") + i*sin(" << ar << ")\n"; </pre>	18. <pre> #include <iostream> using namespace std; int main() { double a = 1.8; int b = 5, c = 2; cout << "a = " << a << ", b = " << b << ", c = " << c << endl; cout << "a + b = " << a + b << endl; cout << "int(a + b) = " << int(a + b) << endl; </pre>

	<pre> return 0; } </pre>		<pre> cout << "int(a) + b = " << int(a) + b << endl; cout << "b / c = " << b / c << endl; cout << "sizeof(b / c) = " << sizeof(b / c) << endl; cout << "(double) b / c = " << (double) b / c << endl; cout << "sizeof((double) b / c) = " << sizeof((double) b / c) << endl; return 0; } </pre>
19.	<pre> #include <cstdio> using namespace std; int main() { int a = 1, b = 100; printf("a = %d, b = %d \n", a, b); a = a + b; b = a - b; a = a - b; printf("a = %d, b = %d\n", a, b); return 0; } </pre>	20.	<pre> #include <cstdio> using namespace std; int main() { int a=1, b = 10, c = 100; printf("a = %4d, b = %4d, c = %4d\n", a, b, c); int s = a + b + c; a = s - a; b = s - b; c = s - c; printf("a = %4d, b = %4d, c = %4d\n", a, b, c); return 0; } </pre>

1.13. Відповіді до задач на читання програмного коду

1. Дано сторони прямокутника, знайти його периметр та площину, або дано 2 числа, знайти їх подвоєну суму та добуток.

2. Дано діаметр кола. Знайти його довжину, радіус і площину.

3. Дано катети прямокутного трикутника. Знайти його гіпотенузу та площину за формулою Герона.

4. Дано координати вершин трикутника. Знайти його сторони.

5. Обмін значеннями змінних із використанням допоміжної змінної.

6. Дано 3 числа a , b , c . Здійснити зсув значень вправо, тобто.

$$a = c, b = a, c = b.$$

7. Дано число a . За 3 дії знайти a^8 .

8. Дано 2-значне число. Обчислити число, цифри якого розташовані в зворотному порядку. Наприклад, $a = 12$, $b = 21$.

9. Дано координати кінців діагоналі прямокутника, сторони якого паралельні осям. Обчислити координати кінців 2-ї діагоналі.

10. Дано кут в радіанах. Обчислити косинус та синус подвоєного кута.
11. Дано сторони трикутника a , b , c . Обчислити кут (в градусах), протилежний стороні c .
12. Дано власна швидкість транспортного засобу, швидкість течії річки, час руху по та проти течії. Знайти пройдений шлях.
13. Дано 1-й член b_1 , знаменник q та n – кількість членів геометричної прогресії. Знайти n -й член та суму n перших членів геометричної прогресії.
14. Дано чисельники та знаменники 2-х дробів. Знайти їх суму, добуток та частку.
15. Знайти $\sqrt{2}$, $\sqrt[3]{3}$, $\sqrt[4]{4}$.
16. Дано число. Знайти результати округлення до меншого цілого, більшого цілого, згідно правил та відкидання дробової частини.
17. Дано дійсна та уявна частини комплексного числа. Знайти його модуль та аргумент.
18. Дано 1 дійсне число a та 2 цілих – b і c . Знайти результати наступних дій: $a + b$, $\text{int}(a + b)$, $\text{int}(a) + b$, b / c , $(\text{double}) b / c$, кількість байт, займаних в пам'яті результатом цілочисельного та звичайного ділення b і c .
19. Обмін значеннями змінних без використання допоміжної змінної.
20. Дано числа a , b , c . Замінити кожне з них сумою 2-х залишившихся.

1.14. Домашнє завдання

[ЕПРЛ], Тема 2 «Лінійні програми»

п.2.10. Задачі на цілочисельне ділення

п.2.18. Задачі на відсотки

Варіанти згідно списку групи.

Тема 2. АЛГОРИТМ РОЗГАЛУЖЕННЯ

ЛАБОРАТОРНА РОБОТА №2

2.1. Умовні вирази. Операції порівняння. Логічні операції

2.2. Умовні конструкції: умовний оператор, оператор switch, тернарний оператор

2.3. Розв'язання задач

2.4. Задачі на написання програмного коду

2.5. Задачі на читання програмного коду

2.6. Відповіді до задач на читання програмного коду

2.7. Домашнє завдання

ТЕОРЕТИЧНИЙ МАТЕРІАЛ:

2.1. Умовні вирази. Операції порівняння. Логічні операції

<https://metanit.com/cpp/tutorial/2.7.php>

2.2. Умовні конструкції: умовний оператор, оператор switch, тернарний оператор

<https://metanit.com/cpp/tutorial/2.12.php>

2.3. Розв'язання задач

1. Дано 2 дійсних числа a , b . Знайти їх максимум

Алгоритм 1 – повний умовний оператор

- 1) Оголосити дійсні змінні a , b й ініціалізувати їх;
- 2) Оголосити дійсну змінну max ;
- 3) Знаходження максимуму 2-х чисел. Використовуючи повний умовний оператор, записати умову

```
якщо     $a > b$ ,
то       $max = a$ ,
інакше   $max = b$ ;
```

4) Вивести на екран значення змінної max

Алгоритм 2 – неповний умовний оператор

- 1) Оголосити дійсні змінні a , b та задати їх значення;
- 2) Оголосити дійсну змінну max й покласти її рівною a ;
- 3) Знаходження максимуму 2-х чисел. Використовуючи неповний умовний оператор, записати умову

```
якщо     $max < b$ ,
то       $max = b$ ,
```

4) Вивести на екран значення змінної max

Алгоритм 3 – тернарний оператор

- 1) Оголосити дійсні змінні a , b й задати їх значення;
- 2) Оголосити дійсну змінну max й, застосовуючи тернарний оператор,

у випадку виконання умови ($a > b$) покласти її значенням a , невиконання – значення змінної b .

3) Вивести на екран значення змінної max

	Алгоритм 1 – Програмний код повний умовний оператор
1	#include <iostream>
2	using namespace std;
3	
4	int main()
5	{
6	// 1) Оголошення дійсних змінних a , b й надання їм значень
7	double $a = 10$, $b = 3$;
8	
9	// 2) Оголошення дійсної змінної max
10	double max ;
11	
12	// 3) Знаходження максимуму 2-х чисел
13	if ($a > b$)
14	{
15	max = a ;
16	}
17	else
18	{
19	max = b ;
20	}
21	
22	// 4) Вивід на екран значення змінної max
23	cout << "max = " << max << endl;
24	
25	return 0;
26	}

	Алгоритм 2 – Програмний код неповний умовний оператор
1	#include <iostream>
2	using namespace std;
3	
4	int main()
5	{
6	// 1) Оголошення дійсних змінних a , b й надання їм значень
7	double $a = 10$, $b = 3$;
8	
9	// 2) Оголошення дійсної змінної max та надання їй значення,
10	// рівного a
11	double $max = a$;
12	
13	// 3) Знаходження максимуму 2-х чисел
14	if ($max < b$)
15	{
16	max = b ;
17	}
18	

19	// 4) Вивід на екран значення змінної max
20	cout << "max = " << max << endl;
21	
22	return 0;
23	}

	Алгоритм 3 – Програмний код тернарний оператор
	<pre>#include <iostream> using namespace std; int main() { // 1) Оголошення змінних a, b й надання їм значень double a = 10, b = 3; // 2) Оголошення змінної max. Знаходження максимуму a і b double max = (a > b) ? a : b; // 3) Вивід на екран значення змінної max cout << "max = " << max << endl; return 0; }</pre>

На екрані:

max = 10

2. Ввести з клавіатури номер місяця (ціле число). Вивести на екран назву пори року. Якщо введене число не може бути номером місяця, вивести на екран повідомлення «Помилковий ввід». Для вводу й виводу використати функції бібліотеки <cstdio>

Алгоритм 1 – вкладені умовні оператори

1) Оголосити цілу змінну month

2) Вивести на екран запрошення «Введіть номер місяця»

3) Ввести з клавіатури значення змінної month

4) Використовуючи вкладені умовні оператори, записати

якщо month < 1 або month > 12

то вивести на екран повідомлення «Введено неприпустиме значення змінної month»,

інакше якщо month = 1 або month = 2 або month = 12,

то вивести на екран повідомлення «Зима»,

інакше якщо month >= 3 та month <= 5,

то вивести на екран повідомлення «Весна»,

інакше якщо month >= 6 та month <= 8,

то вивести на екран повідомлення «Літо»,

інакше вивести на екран повідомлення «Осінь»,

Алгоритм 2 – switch

- 1) Оголосити цілу змінну month
- 2) Вивести на екран запрошення «Введіть номер місяця»
- 3) Ввести з клавіатури значення змінної month
- 4) За допомогою конструкції switch – case – default записати:

у випадках

month $\in \{1, 2, 12\}$, вивести на екран повідомлення «Зима»;

month $\in \{3, 4, 5\}$ – «Весна»;

month $\in \{6, 7, 8\}$ «Літо»;

month $\in \{9, 10, 11\}$ «Осінь».

Якщо значення month $\notin \overline{1, 12}$, вивести на екран повідомлення «Введе-
не неприпустиме значення змінної month».

Алгоритм 1 – Програмний код (вкладені умовні оператори)	
1	#include <stdio>
2	using namespace std;
3	
4	int main()
5	{
6	// 1) Оголошення й введення з клавіатури значення цілої
7	// змінної month
8	int month;
9	printf("Введіть номер місяця\n");
10	scanf("%5d", &month);
11	
12	// 2) З'ясування й вивід на екран пори року
13	if (month < 1 month > 12)
14	printf("Введено неприпустиме значення змінної
15	month\n");
16	else if (month == 1 month == 2 month == 12)
17	printf("Зима\n");
18	else if (month >= 3 && month <= 5)
19	printf("Весна\n");
20	else if (month >= 6 && month <= 8)
21	printf("Літо\n");
22	else
23	printf("Осінь\n");
24	
25	return 0;
26	}

На екрані:

Введіть номер місяця

2

Зима

Алгоритм 2 – Програмний код switch – case – default декільком випадкам (case) відповідає один набір дій

```

1  #include <stdio>
2  using namespace std;
3
4  int main()
5  {
6  // 1) Оголошення й введення з клавіатури значення цілої
7  //    змінної month
8      int month;
9      printf("Введіть номер місяця\n");
10     scanf("%5d", &month);
11
12 // 2) З'ясування й вивід на екран пори року
13     switch (month)
14     {
15         case 1 : case 2 : case 12 :
16             printf("Зима\n");
17             break;
18         case 3 : case 4 : case 5 :
19             printf("Весна\n");
20             break;
21         case 6 : case 7 : case 8 :
22             printf("Літо\n");
23             break;
24         case 9 : case 10 : case 11 :
25             printf("Осінь\n");
26             break;
27         default :
28             printf("Введено неприпустиме значення
29                 змінної month\n");
30             break;
31     }
32
33     return 0;
34
35 }

```

На екрані:

Введіть номер місяця

25

Введено неприпустиме значення змінної month

3. Ввести з клавіатури номер дня тижня (ціле число). Вивести на екран назву дня. Якщо введене число не може бути номером дня, вивести на екран повідомлення «Помилкове введення».

Алгоритм:

- 1) Оголосити цілу змінну day
- 2) Вивести на екран запрошення «Введіть номер дня»
- 3) Ввести з клавіатури значення змінної day
- 4) За допомогою конструкції switch – case – default записати, що у випадку day = 1, вивести на екран «Понеділок», ..., day = 7 – «Неділя», в інших випадках – повідомлення «Помилкове введення»

Програмний код switch - case - default кожному випадку (case) свій набір дій	
1	#include <iostream>
2	using namespace std;
3	
4	int main()
5	{
6	// 1) Оголошення й введення з клавіатури значень змінної day
7	int day;
8	cout << "Введіть номер дня" << endl;
9	cin >> day;
10	
11	// 2) З'ясування й вивід на екран дня тижня
12	switch (day)
13	{
14	case 1 : cout << "Понеділок" << endl; break ;
15	case 2 : cout << "Вівторок" << endl; break ;
16	case 3 : cout << "Середа" << endl; break ;
17	case 4 : cout << "Четвер" << endl; break ;
18	case 5 : cout << "П'ятниця" << endl; break ;
19	case 6 : cout << "Субота" << endl; break ;
20	case 7 : cout << "Неділя" << endl; break ;
21	default : cout << "Помилкове введення\n";
22	break ;
23	}
24	
25	return 0;
	}

На екрані:

Введіть номер дня

2

Вівторок

2.4. Задачі на написання програмного коду

2.4.1. Неповний умовний оператор

1. [Абрамян, If1]. Дано ціле число. Якщо воно додаткове, то додати до нього 1; в іншому випадку не змінювати його. Ввести отримане число на екран.

2. [Абрамян, If4]. Дано три цілих числа. Знайти кількість додаткових чисел у вихідному наборі.

3. [Абрамян, If12]. Дано три числа. Знайти найменше з них.

4. [Абрамян, If20]. На числовій осі розташовані три точки: А, В, С. З'ясувати, яка з двох останніх точок (В або С) розташовані ближче до А, й вивести цю точку й відстань від неї до точки А.

2.4.2. Повний умовний оператор

1. [Абрамян, If2]. Дано ціле число. Якщо воно додаткове, то додати до нього 1; в іншому випадку – відняти від нього 2. Вивести отримане число.

2. [Абрамян, If8]. Дано два нерівних числа. Вивести першим більше з них, а потім – менше.

3. [Абрамян, If17]. Дано три змінні дійсного типу: a, b, c . Якщо їх значення упорядковані за зростання або за спаданням, подвоїти їх; в іншому випадку замінити значення кожної змінної на протилежне. Вивести нові значення змінних a, b, c .

4. [Абрамян, If9]. Дано дві змінні дійсного типу: a, b ($a \neq b$). Перерозподілити значення даних змінних так, щоб в a виявилось менше із значень, а в b — більше. Вивести нові значення змінних a, b .

2.4.3. Вкладені умовні оператори

1. [Абрамян, If3]. Дано ціле число. Якщо воно додатна, додати до нього 1; якщо від'ємне відняти від нього 2; якщо нульове, замінити його на 10. Вивести отримане число

2. [Абрамян, If28]. Дано номер року (додатна ціле число). З'ясувати кількість днів у цьому році, враховуючи, що звичайний рік налічує 365 днів, а високосний — 366 днів. Високосним вважається рік, який ділиться на 4, за винятком тих років, які діляться на 100 та не діляться на 400 (наприклад, роки 300, 1300 та 1900 не є високосними, а 1200 та 2000 — високосні).

3. [Абрамян, If29]. Дано ціле число. Вивести його рядок-опис виду «від'ємне парне число», «нульове число», «додатна непарне число» і т. ін.

4. [Абрамян, If30]. Дано ціле число, яке лежить в діапазоні 1–999. Вивести його рядок-опис виду «парне двозначне число», «непарне тризначне число»

2.4.4. switch – case

1. [Абрамян, Case2]. Дано ціле число K . Вивести рядок-опис оцінки, що відповідає числу K (1 — «погано», 2 — «незадовільно», 3 — «задовільно», 4 — «добре», 5 — «відмінно»). Якщо K не лежить в діапазоні 1–5, то вивести повідомлення «помилка».

2. [Абрамян, Case4]. Дано номер місяця — ціле число в діапазоні 1–12 (1 — січень, 2 — лютий і т.ін.). Визначити кількість днів в цьому місяці для невисокосного року.

3. [Абрамян, Case5]. Арифметичні операції над числами пронумеровані наступним: 1 — складання, 2 — віднімання, 3 — множення, 4 — цілочисельне ділення, 5 — залишок від цілочисельного ділення, 6 — ділення. Дано номер операції N (ціле число в діапазоні 1–6) та цілі числа A і B (B не дорівнює 0). Виконати над числами вказану операцію й вивести на екран результат

4. [Абрамян, Case8-9]. Дано два цілих числа: D (день) і M (місяць), які визначають правильну дату невисокосного року. Вивести значення D і M для дати, яка передує, та наступної за вказаною.

2.5. Задачі на читання програмного коду

1	<code>#include <iostream></code> <code>using namespace std;</code>	2	<code>#include <iostream></code> <code>using namespace std;</code>
---	---	---	---

	<pre> int main() { double x = 5, y = 8; double m = x, n = y; if (x > y) { m = y; n = x; } cout << "m = " << m << " n = " << n << endl; return 0; } </pre>		<pre> int main() { double x = 5, y = 8, z = 4; double n = x; if (y < n) n = y; if (z < n) n = z; cout << "n = " << n << endl; return 0; } </pre>
3	<pre> #include <iostream> using namespace std; int main() { double x = 5, y = 8, z = 4; double s = x + y + z; double p = x * y * z; double a = (s > p) ? s : p; cout << "a = " << a << endl; return 0; } </pre>	4	<pre> #include <iostream> using namespace std; int main() { double a = 15, b = 8, c = 1; cout << "a = " << a << "\t" << "b = " << b << "\t" << "c = " << c << endl; if (a >= b && b >= c) { a *= 2; b *= 2; c *= 2; } else { a = abs(a); b = abs(b); c = abs(c); } cout << "a = " << a << "\t" << "b = " << b << "\t" << "c = " << c << endl; return 0; } </pre>
5	<pre> #include <iostream> using namespace std; int main() { double x = 15, y = 8; cout << "x = " << x << "\t" << "y = " << y << endl; double z = (x > y) ? x - </pre>	6	<pre> #include <iostream> using namespace std; int main() { double x = 15, y = 8; (x > y) ? cout << "x = " << x << endl : cout << "x = " << x << "\t" << "y = " << y << endl; </pre>

	<pre> y : y - x + 1; cout << "z = " << z << endl; return 0; } </pre>		<pre> return 0; } </pre>
7	<pre> #include <iostream> using namespace std; int main() { double x = 2, y = 0, z = 5; cout << "x = " << x << "\t" << "y = " << y << "\t" << "z = " << z << endl; if (x + y + z < 1) { if (x < y && x < z) x = (y + z) / 2; else if (y < x && y < z) y = (x + z) / 2; else if (z < x && z < y) z = (x + y) / 2; } else if (x < y) x = (y + z) / 2; else y = (x + z) / 2; cout << "x = " << x << "\t" << "y = " << y << "\t" << "z = " << z << endl; return 0; } </pre>	8	<pre> #include <iostream> using namespace std; int main() { double a = 2, b = 40, c = 5, d = 6; cout << a << ", " << b << ", " << c << ", " << d << endl; if (a <= b && b <= c && c <= d) { a = d; b = d; c = d; } else if (!(a > b && b > c && c > d)) { a *= a; b *= b; c *= c; d *= d; } cout << a << ", " << b << ", " << c << ", " << d << endl; return 0; } </pre>
9	<pre> #include <iostream> using namespace std; int main() { double x = -2, y = -4; cout << "x = " << x << "\t" << "y = " << y << endl; if (x < 0 && y < 0) { x = -x; y = -y; } else if (x < 0 y < 0) </pre>	10	<pre> #include <iostream> using namespace std; int main() { double x; cin >> x; double fx; if (x <= 0) fx = -x; else if (x <= 1) fx = x; else if (x <= 2) fx = 1; </pre>

	<pre> { x += 0.5; y += 0.5; } else if ((x >= 0 && x < 0.5 x > 2) && (y >= 0 && y < 0.5 y > 2)) { x /= 10; y /= 10; } cout << "x = " << x << "\t" << "y = " << y << endl; return 0; } </pre>		<pre> else fx = 5 - 2*x; cout << "x = " << x << "\t" << "fx = " << fx << endl; return 0; } </pre>
11	<pre> #include <iostream> using namespace std; int main() { int k; cin >> k; switch (k) { case 1 : cout << "пога- но" << endl; break; case 2 : cout << "неза- довільно" << endl; break; case 3 : cout << "задо- вільно" << endl; break; case 4 : cout << "добре" << endl; break; case 5 : cout << "від- мінно" << endl; break; default : cout << "поми- лка" << endl; break; } return 0; } </pre>	12	<pre> #include <iostream> using namespace std; int main() { int n; cin >> n; double a, b; cin >> a >> b; switch (n) { case 1 : cout << a << " + " << b << " = " << a + b << endl; break; case 2 : cout << a << " - " << b << " = " << a - b << endl; break; case 3 : cout << a << " * " << b << " = " << a * b << endl; break; case 4 : if (b == 0) cout << "Divizion by zero \n"; else cout << a << " / " << b << " = " << a / b << endl; break; } return 0; } </pre>

2.6. Відповіді до задач на читання програмного коду

1. Дано 2 числа. Знайти їх максимум та мінімум.
2. Дано 3 числа. Знайти їх мінімум.

3. Дано 3 числа. Знайти максимум їх суми та добутку.
4. Дано дійсні числа a , b і c . Подвоїти ці числа, якщо виконуються нерівності $a \geq b \geq c$; замінити їх абсолютними значеннями, якщо це не так.
5. Дано дійсні числа x , y . Обчислити

$$z = \begin{cases} x - y, & \text{якщо } x > y; \\ y - x + 1, & \text{в іншому випадку.} \end{cases}$$
6. Дано 2 дійсних числа. Вивести на екран перше число, якщо воно більше другого, та обидва числа, якщо це не так.
7. Якщо сума 3 попарно різних дійсних чисел x , y , z менше 1, замінити найменше з них півсумою 2-х других, в іншому випадку замінити менше з x і y півсумою 2-х значень, які залишилися.
8. Дано дійсні числа a , b , c , d . Якщо $a \leq b \leq c \leq d$, то кожне число замінити найбільшим з них; якщо $a > b > c > d$, то залишити незмінними; в іншому випадку – всі числа замінити їх квадратами.
9. Дано дійсні числа x і y . Якщо x і y від'ємні, то кожне з них замінити його модулем; якщо від'ємне тільки одно з них, то обидва значення збільшити на 0.5; якщо обидва значення невід'ємні та жодне з них не належить $[0.5; 2.0]$, обидва значення зменшити в 10 разів; в інших випадках – залишити значення незмінними.
10. Дано дійсне значення x . Обчислить $f(x)$, якщо

для недодатніх	$f(x) = -x;$
$0 < x \leq 1,$	$f(x) = x;$
$1 < x \leq 2,$	$f(x) = 1;$
$2 < x,$	$f(x) = 5 - 2x$
11. Дано ціле число K . Вивести рядок-опис оцінки K (1 — «погано», 2 — «незадовільно», 3 — «задовільно», 4 — «добре», 5 — «відмінно»). Якщо K не належить діапазону 1–5, вивести повідомлення «помилка».
12. Арифметичні дії над числами пронумеровані наступним чином: 1 — складання, 2 — віднімання, 3 — множення, 4 — ділення. Дано номер дії N (ціле число в діапазоні 1–4) та дійсні числа A і B (B не дорівнює 0). Виконайте над числами вказану дію й виведіть результат.

2.7. Домашнє завдання

[ЕПРЛ], Тема 3 «Програми з розгалуженням»

п.3.6. Задачі з повним умовним оператором з складеною умовою

п.3.9. Задачі з неповним умовним оператором

п.3.11. Задачі з вкладеними умовними операторами

п.3.15. Задачі з оператором вибору `switch`

Варіанти згідно списку групи

Тема 3. ЦИКЛІЧНИЙ АЛГОРИТМ

ЛАБОРАТОРНА РОБОТА №3

3.1. Оператор циклу **for**. Оператори **continue**, **break**

3.1.1. Розв'язання задач

3.1.2. Задачі на написання програмного коду

3.1.3. Задачі на читання програмного коду

3.1.4. Відповіді до задач на читання програмного коду

3.1.5. Домашнє завдання

ЛАБОРАТОРНА РОБОТА №4

3.2. Оператори циклу **while**, **do-while**

3.2.1. Розв'язання задач

3.2.2. Стандартний алгоритм Евкліда

3.2.3. Задачі на написання програмного коду

3.2.4. Задачі на читання програмного коду

3.2.5. Відповіді до задач на читання програмного коду

3.2.6. Домашнє завдання

ЛАБОРАТОРНА РОБОТА №5

3.3. Вкладені цикли

3.3.1. Розв'язання задач

3.3.2. Задачі на написання програмного коду

3.3.3. Домашнє завдання

ЛАБОРАТОРНА РОБОТА №3

3.1. Оператор циклу **for**. Оператори **continue**, **break**

ТЕОРЕТИЧНИЙ МАТЕРІАЛ:

<https://metanit.com/cpp/tutorial/2.13.php>

3.1.1. Розв'язання задач

1. [Абрамов, №77 г)]. Дано число n . Обчислити

$$\frac{1}{\sin 1} + \frac{1}{\sin 1 + \sin 2} + \dots + \frac{1}{\sin 1 + \dots + \sin n}$$

Результат – сума дробів.

Знаменник – сума синусів. На k -му кроці циклу до знаменника додається $\sin k$.

У знаменнику останнього дроби n доданків $(\sin 1 + \dots + \sin n)$, отже, цикл виконується n разів. Оскільки кількість кроків циклу відома заздалегідь, застосовується цикл **for**.

Алгоритм:

1) Оголошення змінної цілого типу n та введення її значення з клавіатури

2) Оголошення змінних дійсного типу `summa_drobej` й `znamenatel`, початкове значення яких встановлюється рівним 0 (0 не впливає на результат підсумовування)

3) Цикл `for` з параметром $k = \overline{1, n}$ з кроком зміни параметру, рівним 1.

На кожній ітерації (витку циклу, кроці)) повторюються дії:

– змінна `znamenatel` збільшується на $\sin k$

– до змінної `summa_drobej` додається $1 / \text{znamenatel}$

4) Вивід на екран результату – значення змінної `summa_drobej`

Програмний код оператор циклу <code>for</code> з параметром	
1	<code>#include <iostream></code>
2	<code>#include <cmath></code>
3	<code>using namespace std;</code>
4	
5	<code>int main()</code>
6	<code>{</code>
7	<code> // 1)-2)</code>
8	<code> int n;</code>
9	<code> cout << "n = ";</code>
10	<code> cin >> n;</code>
11	<code> double summa_drobej = 0;</code>
12	<code> double znamenatel = 0;</code>
13	
14	<code> // 3)</code>
15	<code> for (int k = 1; k <= n; k++)</code>
16	<code> {</code>
17	<code> znamenatel += sin(k);</code>
18	<code> summa_drobej += 1 / znamenatel;</code>
19	<code> }</code>
20	
21	<code> // 4)</code>
22	<code> cout << "summa_drobej = " << summa_drobej << endl;</code>
23	
24	<code> return 0;</code>
25	<code>}</code>

На екрані:

`n = 10`

`summa_drobej = 2.83492`

2. [Абрамов, №77 ж)] Дано число n . Обчислити

$$\sqrt{3 + \sqrt{6 + \dots + \sqrt{3(n-1) + \sqrt{3n}}}}$$

На k (кожній) ітерації підкореневий вираз збільшується на $\sqrt{3 * k}$, $k = \overline{1, n}$.

Алгоритм:

- 1) Оголошення й введення з клавіатури значення цілочисельної змінної n ;
- 2) Оголошення дійсної змінної s , початкове значення якої дорівнює 0
- 3) Цикл `for` з параметром $k = \overline{n, 1}$ з кроком зміни параметру, рівним -1 .

На кожній ітерації циклу повторюються дії:

- до змінної s додається $3 * k$
- обчислюється квадратний корінь.

- 4) Вивід на екран значення результуючої змінної s

Програмний код оператор циклу <code>for</code> (з параметром)	
1	<code>#include <iostream></code>
2	<code>#include <cmath></code>
3	<code>using namespace std;</code>
4	
5	<code>int main()</code>
6	<code>{</code>
7	<code> // 1. Вихідне дане - 1)</code>
8	<code> int n;</code>
9	<code> cout << "n = ";</code>
10	<code> cin >> n;</code>
11	
12	<code> // 2. Обчислювальна частина - 2)-3)</code>
13	<code> double s = 0;</code>
14	<code> for (int k = n; k >= 1; k--)</code>
15	<code> {</code>
16	<code> s = sqrt(s + 3*k);</code>
17	<code> }</code>
18	
19	<code> // 3. Вивід результату на екран - 4)</code>
	<code> cout << "s = " << s << endl;</code>
	<code> return 0;</code>
	<code>}</code>

На екрані:

$n = 10$

$s = 2.46993$

3. [Абрамов, №80] Дано дійсне число x . Обчисліть

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \frac{x^{11}}{11!} + \frac{x^{13}}{13!}$$

Результат вивести з 3-мя знаками після коми.

Алгоритм:

- 1) Підключення бібліотеки `<cstdio>`
- 2) Оголошення цілої змінної n й встановлюємо її рівною 13
- 3) Оголошення дійсної змінної x (типа `float`) та введення її зна-

чення з клавіатури

4) Оголошення дійсної змінної `factorial`, початкове значення дорівнює 1

Факторіал – це добуток. Початкове значення добутку дорівнює 1, так як множення на 1 не впливає на результат

5) Оголошення дійсної змінної `summa`, початкове значення дорівнює x

6) Оголошення цілої змінної `znak`, початкове значення дорівнює -1

7) Цикл `for` з параметром $k = \overline{3, n}$ з кроком зміни параметру, рівним 2.

На кожній ітерації повторюються дії:

– змінна `znak` змінює значення на протилежне (1 на -1 , -1 на 1)

– змінна `factorial` множиться на $(k - 1) * k$

– змінна `summa` збільшується на $znak * x^k / factorial$

8) Вивід на екран значення результуючої змінної `summa` с 3 знаками після коми

Програмний код оператор циклу <code>for</code> (з параметром)	
1	<code>#include <stdio></code>
2	<code>#include <cmath></code>
3	<code>using namespace std;</code>
4	
5	<code>int main()</code>
6	<code>{</code>
7	<code> // 1. Вхідні дані</code>
8	<code> int n = 13;</code>
9	<code> float x;</code>
10	<code> printf("x = ");</code>
11	<code> scanf("%f", &x);</code>
12	
13	<code> // 2. Обчислювальна частина</code>
14	<code> double factorial = 1;</code>
15	<code> int znak = 1;</code>
16	<code> double summa = x;</code>
17	<code> for (int k = 3; k <= n; k += 2)</code>
18	<code> {</code>
19	<code> znak = -znak;</code>
20	<code> factorial *= (k - 1) * k;</code>
21	<code> summa += znak * pow(x, k) / factorial;</code>
22	<code> }</code>
23	
24	<code> // 3. Вивід результату на екран</code>
25	<code> printf("summa = %5.3g \n", summa);</code>
26	
27	<code> return 0;</code>
28	<code>}</code>

На екрані:

`x = 2`

`summa = 0.909`

4. Знайти номер співмножника, на якому добуток

$$1 \cdot 3 \cdot 5 \cdot 7 \cdot \dots$$

перевищить 10^{11} .

Алгоритм:

1) В розділі `const` визначити константу `MAX_PRODUCT` типа `long` й покласти її рівною 10^{11} .

`MAX_PRODUCT` визначена у розділі `const`, тому що. її значення у програмі не змінюється.

Тип `int` займає в пам'яті (як правило) 4 байта, й має діапазон:

$$-2\,147\,483\,648 \dots 2\,147\,483\,647$$

10^{10} належить цій множини, а 10^{11} вже ні. Тому константа віднесена до типу `long`, який має діапазон

$$-9\,223\,372\,036\,854\,775\,808 \dots +9\,223\,372\,036\,854\,775\,807.$$

2) Оголосити змінну `product` типа `long`, початковим значенням якої є 1.

3) Оголосити цілочисельну змінну `k` – номер співмножника – й покласти її рівною 0

Змінні доступні в межах тільки того блоку кода, в якому вони визначені. `k` – результуюча змінна, її значення повинно бути відомо після закінчення роботи циклу, тому її потрібно оголосити до циклу.

4) В заголовку циклу `for` 2-я частина – умова продовження роботи:

$$\text{product} \leq \text{MAX_PRODUCT}$$

5) Дії, які повторюються в циклі:

– збільшити на 1 значення змінної `k`

– помножити `product` на `k`-е непарне число

$$2 \cdot k - 1, \quad k = 1, 2, 3, \dots \text{ – непарні числа}$$

6) вивід на екран значення змінної `k`

Програмний код оператор циклу <code>for</code>	
1	<code>#include <iostream></code>
2	<code>#include <cmath></code>
3	<code>using namespace std;</code>
4	
5	<code>int main()</code>
6	
7	
8	<code>{</code>
9	<code> const long MAX_PRODUCT = pow(10, 11); // 1)</code>
10	
11	<code> long product = 1; // 2)</code>
12	<code> int k = 0; // 3)</code>
13	<code> for (; product <= MAX_PRODUCT;) // 4)</code>
14	<code> {</code>

15	// 5)
16	k++;
17	product += 2*k - 1;
18	}
19	
20	cout << "k = " << k << endl; // 6)
21	return 0;
	}

На екрані:

k = 12

3.1.2. Задачі на написання програмного коду

1. [Абрамян, For10]. Дано ціле число $N (> 0)$. Знайти суму

$$1 + 1/2 + 1/3 + \dots + 1/N$$

2. [Абрамян, For13]. Дано ціле число $N (> 0)$. Знайти значення виразу $1.1 - 1.2 + 1.3 \cdot \dots$ (N доданків). Умовний оператор не використовувати.

3. [Абрамян, For33]. Дано ціле число $N (> 1)$. Послідовність чисел Фібоначчі F_K (цілого типу) визначається наступним чином:

$$F_1 = 1, F_2 = 1, F_K = F_{K-2} + F_{K-1}, K = 3, 4, \dots$$

Вивести елементи $F_1, F_2, \dots, F_N$.

У наступних задачах перед розв'язанням задачі розібрати приклад, наведений на початку параграфа.

4. [ЕПРЛ, п.4.3, Вар.17, №1]. Надрукуйте всі двохзначні та трьохзначні числа, які складаються з різних цифр.

5. [ЕПРЛ, п.4.3, Вар.17, №2]. Знайти всі трьохзначні числа, що дорівнюють квадрату якогось двозначного числа.

6. [ЕПРЛ, п.4.4, Вар.17, №1]. Знайдіть кількість та середнє арифметичне натуральних 5-значних чисел, у яких перша цифра дорівнює останній, а друга – передостанній (наприклад, 12421, 47174).

7. [ЕПРЛ, п.4.4, Вар.17, №2]. Ввести з клавіатури n будь-яких чисел. Знайти середнє арифметичне всіх чисел, з парними порядковими номерами

8. [ЕПРЛ, п.4.5, Вар.17, №1]. Дано n цілих чисел. Чи всі вони належать інтервалу $[x, y]$?

9. [ЕПРЛ, п.4.5, Вар.17, №2] Дано n цілих чисел. Чи є в цій послідовності тризначні додатні числа, які складаються з різних цифр?

10. [ЕПРЛ, п.4.5, Вар.17]. Дано ціле число t . Надрукувати з 3 знаками після крапки n чисел

$$a_i = \frac{\sqrt{5i+8}}{i^2+1}, \quad i = \overline{1, n}$$

11. [ЕПРЛ, п.4.8, Вар.17]. Знайдіть значення функції на вказаному інтервалі з вказаним кроком:

$$f(x) = \begin{cases} \frac{\sqrt{x} + 1}{x^2 + 2}, & 0 \leq x \leq 10; \\ \frac{|2 + x| + 1}{3x}, & \text{у ост. вип.} \end{cases}$$

$x \in [-10, 10]$, $h = 2,5$.

13. [ЕПРЛ, п.4.9, Вар.17, №1]. Дано n цілих чисел. Знайти серед цих чисел перше двозначне число, яке складається з різних цифр та його порядковий номер. Якщо таких чисел немає, то вивести NO.

14. [ЕПРЛ, п.4.9, Вар.17, №2]. Дано n цілих чисел. Знайти серед цих чисел останнє число, яке належить інтервалу $[x, y]$ та кратне b та його порядковий номер. Якщо таких чисел немає, то вивести NO.

15. [ЕПРЛ, п.4.10, Вар.17, №1]. Ввести з клавіатури n будь-яких чисел та число k . Знайдіть серед них число (його порядковий номер і значення) найближче до k . Якщо таких чисел декілька, то знайдіть перше.

16. [ЕПРЛ, п.4.10, Вар.17, №2]. Ввести з клавіатури n будь-яких чисел. Знайти серед них найменше тризначне число та найбільше двозначне число. Якщо є декілька таких чисел, то визначте порядкові номери останніх з них. Якщо немає таких чисел, то вивести повідомлення.

3.1.3. Задачі на читання програмного коду

1	<pre>#include <iostream> using namespace std; int main() { int c1, c2; for (int i=10; i<=99; i++) { c1 = i / 10; c2 = i % 10; if (c1 != c2) { cout << i << endl; } } return 0; }</pre>	2	<pre>#include <iostream> using namespace std; int main() { int c; for (int k=21; k<=89; k +=2) { c = k / 10; if (c % 2 == 0) { cout << k << endl; } } return 0; }</pre>
3	<pre>#include <iostream> using namespace std; int main() { int c1, c2, c3, c4, c; int s = 0; int n = 0; for (int k=1000; k<=9999; k++) { c = k;</pre>	4	<pre>#include <iostream> using namespace std; int main() { int n = 10; int c; int k = 0; for (int i = 1; i <= n; i++) {</pre>

	<pre> c1 = c / 1000; c = c % 1000; c2 = c / 100; c = c % 100; c3 = c / 10; c4 = c % 10; if (c1 == c4 && c2 == c3) { s += k; n++; } } double sa = (double) s / n; cout << sa << endl; return 0; } </pre>		<pre> cin >> c; if (100 <= c && c <= 999) { k++; } } if (k == 0) cout << "no" << endl; else cout << k << endl; return 0; } </pre>
5	<pre> #include <iostream> using namespace std; int main() { int n = 10; int a; bool ap = true; for (int i=1; i <= n; i++) { cin >> a; ap = ap && (a > 0); if (ap == false) { break; } } (ap == true) ? cout << "yes" << endl : cout << "no" << endl; return 0; } </pre>	6	<pre> #include <cstdio> #include <cmath> using namespace std; int main() { int n = 10; double a; for (int k=1; k <= n; k++) { a = 3. / (k*k + 1) + pow(0.15, 1./k); printf("a[%d] = %5.3f\n", k, a); } return 0; } </pre>
7	<pre> #include <cstdio> #include <cmath> using namespace std; int main() { double a = -11, b = 11, h = 2.75; double fx; for (double x = a; x <= b; x += h) </pre>	8	<pre> #include <iostream> using namespace std; int main() { int n = 10; double x = -10, y = 15; double t; int f, nf; bool v = false; for (int i = 1; i <= n; </pre>

	<pre> { fx = (-7<=x && x<=7) ? 25 / sqrt(x + 9) : (4*x*x + 1) / abs(x); printf("f(%5.3f) = %5.3f\n", x, fx); } return 0; } </pre>		<pre> i++) { cin >> t; if (x <= t && t <= y) { f = t; nf = i; v = true; break; } } if (v == true) { cout << f << "\t" << nf << endl; } else { cout << "no\n" << endl; } return 0; } </pre>
9	<pre> #include <iostream> #include <cstdlib> #include <ctime> using namespace std; int main() { srand(time(0)); int n = 100; int a = -1000, b = 1050; int l, nl; bool v = false; int t, c1, c2, c3; for (int i = 1; i <= n; i++) { t = a + rand() % (b- a+1); if (100 <= t && t <= 999 && t % 11 == 0) { l = t; nl = i; v = true; } } (v == true) ? cout << l << "\t" << </pre>	10	<pre> #include <iostream> using namespace std; int main() { int n = 10; double k = 6.5; double x; cin >> x; double xr = abs(x - k); double l = x, lr = xr; int ln = 1; for (int i = 2; i <= n; i++) { cin >> x; xr = abs(x - k); if (lr >= xr) { l = x; lr = xr; ln = i; } } cout << l << "\t" << ln << endl; return 0; } </pre>

	<pre> nl << endl : cout << "no\n" << endl; return 0; } </pre>		
11	<pre> #include <iostream> #include <cstdlib> #include <ctime> using namespace std; int main() { srand(time(0)); int n = 10; double a = -10, b = 20; double x; int k = 0; for (int i = 1; i <= n; i++) { x = a + (b - a) * rand() / RAND_MAX; if (x == int(x)) { k++; } } cout << k << endl; return 0; } </pre>	12	<pre> #include <iostream> using namespace std; int main() { int n = 10; double x1, x2; cin >> x1 >> x2; double q = x2 / x1; double q1; x1 = x2; bool gp = true; for (int i = 3; i <= n; i++) { cin >> x2; q1 = x2 / x1; if (q1 != q) { gp = false; break; } x1 = x2; } if (gp) cout << "yes" << endl; else cout << "no" << endl; return 0; } </pre>

3.1.4. Відповіді до задач на читання програмного коду

1. Надрукуйте всі 2-значні числа, які складаються з різних цифр
2. Надрукуйте всі 2-значні непарні числа, перша цифра яких парна.
3. Знайдіть середнє арифметичне 4-значних чисел, 1-а цифра яких дорівнює 4-й, 2-а – 3-й.
4. З клавіатури введені n цілих чисел. Знайдіть кількість 3-значних чисел серед них.
5. З клавіатури введені n цілих чисел. Чи є всі вони додатними?
6. Дано ціле число n . Вивести на екран з 3-ма знаками після коми n членів послідовності, заданої формулою:

$$a_k = \frac{3}{k^2+1} + \sqrt[k]{0,15k}, \quad k = \overline{1, n}$$

7. Протабулювати на відрізку $[-11, 11]$ з кроком $h = 2.75$ функцію

$$f(x) = \frac{\frac{25}{\sqrt{x+9}}}{\frac{4x^2+1}{|x|}}, \quad -7 \leq x \leq 7;$$

. в ост. вип.

8. Дано n цілих чисел та дійсні числа x, y . Знайти серед них 1-е число, яке належить відрізку $[x, y]$ та його порядковий номер. Якщо таких чисел немає, вивести на екран "no".

9. Дано цілі числа a, b ($b \geq 1000$) та n – кількість випадкових чисел з $\{a, a+1, \dots, b\}$. Знайти останнє 3-значне число, яке ділиться на 11, та його номер. Якщо таких чисел немає, вивести на екран "no".

10. Дано ціле число n , дійсне число k . З клавіатури ввести n дійсних чисел, знайти серед них найближче до k число та його порядковий номер. Якщо таких чисел декілька, знайти останнє.

11. Дано n випадкових дійсних чисел з відрізка $[-10, 20]$. Знайти кількість цілих чисел серед них.

12. Ввести з клавіатури n дійсних чисел. Чи утворюють вони геометричну прогресію?

3.1.5. Домашнє завдання

[ЕПРЛ], Тема 4 «Цикли з параметром»

п.4.3. Задачі на пошук в інтервалі чисел, які задовольняють деякій умові

п.4.4. Задачі на знаходження кількості, суми, добутку, середнього арифметичного

п.4.5. Задачі з використання логічних змінних

п.4.7. Задачі на обчислення елементів послідовності за їх порядковим номером

п.4.8. Задачі на створення таблиць значень функцій

п.4.9. Задачі на пошук першого та останнього чисел з заданою властивістю

п.4.10. Задачі на пошук максимального та мінімального

п.4.12. Задачі з випадковими числами

п.4.13. Задачі з використанням 2-х змінних для вводу послідовності чисел

п.4.14. Задачі на обчислення елементів послідовності через попередні елементи

Варіанти згідно списку групи

ЛАБОРАТОРНА РОБОТА №4

3.2. Оператори циклу `while`, `do-while`.

ТЕОРЕТИЧНИЙ МАТЕРІАЛ:

<https://metanit.com/cpp/tutorial/2.13.php>

3.2.1. Розв'язання задач

1. [Абрамян, While3]. Дано цілі додатні числа N і K . Використовуючи тільки операції складання та віднімання, знайти частку від цілочисельного ділення N на K , а також залишок від цього ділення.

Алгоритм

- 1) Оголосити й ініціалізувати цілі змінні n і k
- 2) Оголосити цілу змінну q (частку) та r (залишок від ділення), покласти q рівним 0, r рівним n .
- 3) Ідея рішення: оскільки $n = k \cdot q + r$, то $r = n - k - \dots - k$, k віднімається від n q разів). Процес віднімання зупиняється, коли r стає меншим k . Кількість віднімань q невідомо заздалегідь, тому для розв'язання задачі використовується цикл з передумовою $r \geq k$.

Дії, які повторюються в циклі:

- а) зменшення залишку від ділення r на k
- б) збільшення частки q на 1
- 4) Вивід на екран q і r

Програмний код оператор циклу з передумовою while	
1	#include <iostream>
2	using namespace std;
3	
4	int main()
5	{
6	// 1. Вихідні дані
7	int n = 100, k = 12;
8	
9	// 2. Знаходження частки та залишка від ділення n на k
10	int q = 0;
11	int r = n;
12	while (r >= k)
13	{
14	r -= k;
15	q++;
16	}
17	
18	// 3. Вивід на екран
19	cout << "q = " << q << "\t" << "r = " << r << endl;
20	
21	return 0;
22	}

На екрані:

q = 8 r = 4

2. [Абрамов, №866)]. Дано натуральне число n . Знайти суму його цифр.

Алгоритм:

- 1) Оголосити й задати значення цілій змінній n

2) Оголосити цілі змінні `nn`, `digit`, `sum_digits` та покласти `nn` рівною `n`, `sum_digits` рівною 0

3) Кількість цифр в числі заздалегідь невідома, тому організувати цикл `while` з передумовою (`nn != 0`)

Дії, які повторюються в циклі:

а) знайти останню цифру числа `nn`, дорівнює залишку від ділення на 10

б) додати знайдену цифру до суми цифр

в) розділити `nn` на 10

4) Вивід значення змінної `sum_digits` на екран

Програмний код оператор циклу з передумовою <code>while</code>	
1	<code>#include <iostream></code>
2	<code>using namespace std;</code>
3	
4	<code>int main()</code>
5	<code>{</code>
6	<code> // 1. Вихідне дане</code>
7	<code> int n = 7651;</code>
8	
9	<code> // 2. Обчислення суми цифр числа n</code>
10	<code> int nn = n;</code>
11	<code> int digit;</code>
12	<code> int sum_digits = 0;</code>
13	<code> while (nn != 0)</code>
14	<code> {</code>
15	<code> digit = nn % 10;</code>
16	<code> sum_digits += digit;</code>
17	<code> nn /= 10;</code>
18	<code> }</code>
19	
20	<code> // 3. Вивід на екран суми цифр числа n</code>
21	<code> cout << "sum_digits = " << sum_digits << endl;</code>
22	
23	<code> return 0;</code>
24	<code>}</code>

На екрані:

```
sum_digits = 19
```

3.2.2. Алгоритм Евкліда обчислення НСД(a, b)

https://uk.wikipedia.org/wiki/Алгоритм_Евкліда

Програмний код оператор циклу з передумовою <code>while</code>	
1	<code>#include <iostream></code>
2	<code>using namespace std;</code>
3	
4	<code>int main()</code>
5	<code>{</code>
6	<code> // 1. Вихідні дані</code>
7	<code> int a = 900, b = 280;</code>

```

8
9      // 2. Алгоритм Евкліда
10     int m = a, n = b;
11     while ( m != n )
12     {
13         if (m > n)
14         {
15             m -= n;
16         }
17         else
18         {
19             n -= m;
20         }
21     }
22
23     // Вивід НСД на екран
24     cout << "НСД(" << a << ", " << b << ") = " << m << endl;
25
26     return 0;
27 }

```

На екрані:

НСД(900,280) = 20

3.2.3. Задачі на написання програмного коду

1. [Абрамов, №88б)]. Дано натуральне число n . Знайти число, яке має зворотний порядок цифр.
2. Дано натуральне число n . Вивести на екран цифри числа n в прямому порядку.
3. [Абрамов, №88в)]. Дано натуральне число n . Переставити першу та останню цифри числа.
4. [Абрамов, №88г)]. Дано натуральне число n . Приписати 1 в початок та кінець числа.
5. [Абрамян, While24]. Дано натуральне число n . Перевірити, чи є n членом послідовності Фібоначчі.
6. [Абрамян, While15]. Початковий вклад у банку дорівнює 1000 грн. В кінці кожного місяця розмір вкладу збільшується на p процентів від наявної суми (p — дійсне число, $0 < p < 25$). За даним p визначити, через скільки місяців розмір вкладу перевищить 1100 грн., вивести знайдену кількість місяців k (ціле число) та підсумковий розмір вкладу s (дійсне число).
7. [Абрамян, While28]. Дано дійсне число ε (> 0). Послідовність дійсних чисел a_k визначається наступним чином: $a_1 = 2$, $a_k = 2 + 1/a_{k-1}$, $k = 2, 3, \dots$. Знайти перший з номерів k , для яких виконується умова $|a_k - a_{k-1}| < \varepsilon$, та вивести цей номер, а також числа a_{k-1} і a_k .
8. Знайти всі правильні нескоротні дробі, знаменник яких не перевищує 10.

9. Дано натуральні числа m і n . Знайти в множині $\{1, \dots, n\}$ всі числа, взаємно прості з m .

10. [Абрамян, While28]. Дано ціле число n (> 1). Вивести найменше з цілих чисел k , для яких сума $1 + 1/2 + \dots + 1/k$ буде більше або рівна n , та саму цю суму.

3.2.4. Задачі на читання програмного коду

1	<pre>#include <iostream> #include <cmath> using namespace std; int main() { int n = 1000; int k = 0; while (pow(3,k) < n) { k++; } if (k != 0) { k--; } cout << "k = " << k << endl; return 0; }</pre>	2	<pre>#include <iostream> using namespace std; int main() { int n = 81; int nn = n; while (nn % 3 == 0) { nn /= 3; } if (nn == 1) { cout << "true" << endl; } else { cout << "false" << endl; } return 0; }</pre>
3	<pre>#include <iostream> using namespace std; int main() { double w = 0.3; long int k = 0; double s = 0; int z = 1; while (s <= w) { k++; z = -z; s += z * (double) k / (k+1); } cout << "k = " << k << endl; }</pre>	4	<pre>#include <iostream> using namespace std; int main() { int z; cin >> z; while (z % 2 == 0) { cout << "ERROR" << endl; cin >> z; } cout << "z = " << z << endl; return 0; }</pre>

	<pre> return 0; } </pre>		
5	<pre> #include <iostream> using namespace std; int main() { int n = 0; int m1 = n % 10; int m2 = m1; int d; int nn = n / 10; while (nn != 0) { d = nn % 10; if (d > m1) m1 = d; else if (d < m2) m2 = d; nn /= 10; } cout << "m1 = " << m1 << endl; cout << "m2 = " << m2 << endl; return 0; } </pre>	6	<pre> #include <iostream> using namespace std; int main() { int n = 44144; int d = n % 10; int nn = n / 10; int dd; while (nn != 0) { dd = nn % 10; if (dd != d) break; nn /= 10; } (nn == 0) ? cout << "true" << endl : cout << "false" << endl; return 0; } </pre>

3.2.5. Відповіді до задач на читання програмного коду

1. Дано натуральне число n . Знайти найбільше ціле число k , при якому виконується нерівність $3^k < n$.
2. Дано натуральне число n . Якщо воно є ступенем числа 3, вивести true, якщо ні — вивести false.
3. Дано додатна дійсне число w ($w \leq 1.8$).. Знайти номер доданка, при якому сума знакозмінного ряду $-1/2 + 2/3 - 3/4 + \dots$ перевищує w .
4. Користувач повинен ввести непарне число. У випадку вводу парного числа на екран повинно виводитися повідомлення про помилку, після чого дії повинні повторюватися до введення правильного значення.
5. Дано ціле число. Знайти його найбільшу й найменшу цифри.
6. Якщо задане число складається з однакових цифр, вивести на екран true, з різних – false.

3.2.6. Домашнє завдання

[ЕПРЛ], Тема 5 «Цикли з умовою»

п.5.2. Пошук декількох чисел, що задовольняють деякій умові

п.5.3. Задачі на перебір цифр натурального числа

п.5.4. Задачі на зміну натурального числа

п.5.5. Задачі на алгоритм Евкліда

п.5.6. Задачі на обчислення елементів послідовності за заданою точністю

Варіанти згідно списку групи

ЛАБОРАТОРНА РОБОТА №5

3.3. Вкладені цикли

ТЕОРЕТИЧНИЙ МАТЕРІАЛ:

<https://metanit.com/cpp/tutorial/2.13.php>

<https://purecodecpp.com/archives/770>

<https://www.youtube.com/watch?v=HjaCTX4WAmI>

3.3.1. Розв'язання задач

1. Намалювати символом «зірочка» (*) прямокутник 3 x 5 (3 рядка по 5 зірочок у кожному)

Алгоритм:

1) Оголосити та задати значення цілим змінним $m = 3$ (кількість рядків) і $n = 5$ (кількість зірочок в рядку)

2) Зовнішній цикл по номером рядка i , внутрішній – по номеру j зірочки в рядку. Після виводу на екран рядка зірочок, перевід рядку

Програмний код Вкладені цикли	
1	#include <iostream>
2	using namespace std;
3	
4	int main()
5	{
6	int m = 3, n = 5;
7	
8	for (int i = 1; i <= m; i++)
9	{
10	for (int j = 1; j <= n; j++)
11	{
12	cout << "*";
13	}
14	cout << endl;
15	}
16	
17	return 0;
18	} та

На екрані:

```
*****
*****
*****
```

2. Намалювати зірочками нижній правий трикутник

Алгоритм

1) Оголосити та задати значення цілої змінної $n = 3$ (кількість рядків)

- 2) Зовнішній цикл – по номеру рядка
- 3) 1-й внутрішній цикл виводить на екран $n-i$ пробілів, 2-й – i зірочок, після чого здійснюється перевід рядку.

Програмний код Вкладені цикли	
1	#include <iostream>
2	using namespace std;
3	
4	int main()
5	{
6	int n = 5;
7	
8	for (int i = 1; i <= n; i++)
9	{
10	for (int j = 1; j <= n-i; j++)
11	{
12	cout << " ";
13	}
14	for (int j = 1; j <= i; j++)
15	{
16	cout << "*";
17	}
18	cout << endl;
19	}
20	
21	return 0;
22	}

На екрані:

```

      *
     **
    ***
   ****
  *****

```

3. Дано натуральні числа a , b . Виведіть на екран числа з множини $\{a, a+1, \dots, b\}$, які мають 4 дільника.

Алгоритм:

- 1) Оголосити, задати значення цілих змінних a , b
- 2) Оголосити цілі змінні d (кількість дільників числа), p (половина числа)
- 3) Зовнішній цикл for з параметром i , який змінюється від a до b .
 p дорівнює числу, цілочисельно поділеному на 2.
 d дорівнює 2, тому що 2 дільника (1 й само число), у заданого числа завжди є.

Оскільки всі дільники числа, окрім 1 та самого числа, належать множині $\{2, \dots, p\}$, у внутрішньому циклі, який обчислює кількість дільників числа i , параметр $j = 2, \dots, p$.

Перевірка: якщо кількість дільників числа дорівнює 4, то число і виводиться на екран.

Програмний код Вкладені цикли	
1	#include <iostream>
2	using namespace std;
3	
4	int main()
5	{
6	int a = 10, b = 25;
7	
8	int d, p;
9	for (int i = a; i <= b; i++)
10	{
11	p = i / 2;
12	d = 2;
13	for (int j = 2; j <= p; j++)
14	{
15	if (i % j == 0)
16	{
17	d++;
18	}
19	}
20	if (d == 4)
21	{
22	cout << i << "\t";
23	}
24	}
25	return 0;
	}

На екрані:

10 14 15 21 22

3.3.2. Задачі на написання програмного коду

1. Намалювати зірочками фігури:

***** ***** *****	***** ***** *****	* ** *** **** *****	***** ***** *** ** *
***** **** *** ** *	* *** ***** *** *	***** *** * * *** *****	* * * * * * * * * * * *

2. Дано натуральні числа a , b . Вивести на екран числа з множини $\{a, a+1, \dots, b\}$, які є квадратами простих чисел.

3. Дано натуральні числа a , b . Знайти число з найбільшою кількістю ді-

льніків. Якщо таких чисел декілька, вивести на екран найменше з них

4. Дано число n . З n членів послідовності Фібоначчі вивести на екран прості числа.

3.3.3. Домашнє завдання

[ЕПРЛ], Тема 4 «Цикли з параметром»

п.4.15. Задачі на вивід чисел у вигляді таблиці

п.4.16. Задачі на перебір варіантів

п.4.17. Задачі на знаходження дільників натуральних чисел

[ЕПРЛ], Тема 5 «Цикли з умовою»

п.5.7. Задачі з вкладеними циклами

Варіанти згідно списку групи