**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра програмних систем і технологій**

**О. С. Бичков**

**Методичні вказівки до лабораторних занять з дисципліни**

**«Вступ до об’єктно-орієнтованого програмування.   
Мова С# і платформа MS.Net Framework »**

для студентів спеціальності

121 «Інженерія програмного забезпечення»

освітнього рівня «бакалавр»

Київ 2021

# Розділ 1. Введення в мову програмування C#

## Лабораторна робота № 1 Мова C#. Методи введення, виведення, перетворення типів, оператори: умовний, вибору, циклу

### Мета роботи:

1. Прості обчислення з перетворенням типів даних. Метод Parse

2. Прості арифметичні вирази

3. Клас Convert

4. Перетворення з арифметичного типу в тип string. Метод ToString

### Методичні рекомендації до виконання лабораторної роботи

1. Прочитайте лекцію.

2. Прочитайте цю лабораторну роботу та виконайте наведені в ній приклади (вони всі працездатні)

3. Зверніть увагу на те, що при введенні чисел з консолі у змінні арифметичного типу потрібно застосовувати явне приведення типу, так як неявного перетворення з типу **string** до арифметичного немає.

4. При виконанні завдань зверніть увагу на оператор **switch** (перемикач). В С# тип виразу в цьому операторі може бути не тільки цілочисельним, а і типу char або string.

### Порядок виконання роботи

1. Створити директорію Lab1, в якій буде розміщуватися проект цієї лабораторної роботи.

2. Виконати завдання свого варіанту у вигляді одного проекту

### Приклади виконання завдань

#### 1. Прості обчислення з перетворенням типів даних. Метод Parse

Розглянемо приклад створення простого консольного застосунку, який виконує прості арифметичні обчислення. Запустимо Visual Studio, виберемо тип проекту Console Application, дамо йому назву **Lab1\_1** і натиснемо кнопку **OK**.

Створимо програму, яка вводить два цілі числа, знаходить їх суму і виводить результат на консоль. Потім вводить два дійсні числа, обчислює їх частку і також виводить на консоль.

Ось таким може бути код цієї програми. Його можна просто скопіювати в VS і виконати.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Lab2\_1

{

class Program

{

static void Main(string[] args)

{

//Знаходження суми цілих чисел

Console.WriteLine("Введіть перше число");

int a1 = int.Parse(Console.ReadLine());

Console.WriteLine("Введіть друге число");

int a2 = int.Parse(Console.ReadLine());

Console.WriteLine("Сума чисел a1+a2= " + (a1 + a2));

//Знаходження частки дійсних чисел

Console.WriteLine("Введіть перше число");

double d1 = double.Parse(Console.ReadLine());

Console.WriteLine("Введіть друге число");

double d2 = double.Parse(Console.ReadLine());

Console.WriteLine("Частка чисел d1/d2= " + (d1/d2));

Console.ReadKey();

}

}

}

Дані, які вводяться з консолі мають тип string (це рядки тексту). Тому введені числа потрібно перетворити в арифметичний тип. В нашому прикладі виконується явне перетворення типів з типу string в типи **int** і **double** з використанням методу Parse і неявне - при виведенні результатів обчислень на консоль.

Розглянемо ще приклад явного перетворення типу.

**Приклад**

string str = Console.ReadLine();

int i = int.Parse(str);

float f = float.Parse(str);

double d = double.Parse(str);

#### 2. Прості арифметичні вирази

Розглянемо дещо складніший приклад обчислення арифметичного виразу. Нехай нам потрібно обчислити значення поліному р=ax3+bx2 + cx+d для будь-якого дійсного x (вводиться з консолі). Значення a,b,c,d є цілими і ініціалізуються в коді. Текст програми може бути таким:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Lab2\_2

{

class Program

{

static void Main(string[] args)

{

//обчислити значення поліному р=ax^3+bx^2+cx+d

int a = 100;

int b = 20;

int c = 25;

Console.WriteLine("Введіть x");

double x = double.Parse(Console.ReadLine());

double p = a \* x \* x \* x + b \* x \* x + c \* x + 30;

Console.WriteLine("p = " + p);

Console.ReadKey();

}

}

}

#### 3. Клас Convert

Клас Convert містить статичні методи, які також можна використовувати для перетворень типів. Розглянемо приклад використання цього класу.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Lab1\_2

{

class Program

{

/// <summary>

/// Тестування методів класу Convert

/// </summary>

static void Main(string[] args)

{

string s;

byte b;

int n;

double x;

bool flag;

char sym;

DateTime dt;

sym = '7';

s = Convert.ToString(sym);

x = Convert.ToDouble(s);

n = Convert.ToInt32(x);

b = Convert.ToByte(n);

flag = Convert.ToBoolean(b);

x = Convert.ToDouble(flag);

s = Convert.ToString(flag);

s = "300";

n = Convert.ToInt32(s);

s = "14.09";

s = "14.09.2008";

dt = Convert.ToDateTime(s);

}

}

}

#### 4. Перетворення з арифметичного типу в тип string. Метод ToString

Так як метод ToString() визначений в базовому класі object, він має свої реалізації для всіх арифметичних типів даних. Далі наведено простий приклад з лекції 2, який демонструє застосування цього методу.

В першому варіанті використовується **явне** перетворення типу з арифметичного до типу string, а у другому - **неявне**.

Варіант 1.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Lab2\_3

{

class Program

{

/// <summary>

/// Демонстрація перетворення в рядок

/// даних різного типа.

/// </summary>

static void Main(string[] args)

{

string name;

int age;

double salary;

name = "Василь Іванов";

age = 20;

salary = 2700;

string s = "Ім'я: " + name +

". Вік: " + age.ToString() +

". Зарплата: " + salary.ToString();

Console.WriteLine(s);

Console.ReadKey();

}

}

}

Варіант 2

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Lab2\_3

{

class Program

{

/// <summary>

/// Демонстрація перетворення в рядок

/// даних різного типа.

/// </summary>

static void Main(string[] args)

{

string name;

int age;

double salary;

name = "Василь Іванов";

age = 20;

salary = 2700;

string s = "Ім'я: " + name +

". Вік: " + age +

". Зарплата: " + salary;

Console.WriteLine(s);

Console.ReadKey();

}

}

}

#### 5. Створення консольних проектів з використанням операторів if

Створимо консольний проект для організації діалогу таким чином, щоб реалізувати наведений нижче алгоритм.

* Вивести на консоль запрошення для введення імені.
* Ввести своє ім'я і зберегти в текстовому рядку: string myName.
* Якщо нічого не введено, то вивести повідомлення про це і завершити роботу.
* Якщо щось введено, то вивести рядок привітання.
* Вивести рядок із запитом віку
* Якщо нічого не введено, то вивести повідомлення про це і завершити роботу.
* Якщо вік введено, привести число до цілого (int myAge).
* Якщо число myAge<15, вивести повідомлення "Ви ще не студент. "
* Якщо число myAge>40, вивести повідомлення "Вчитися ніколи не пізно!"

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleHello1

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введіть ім'я");

string myName = Console.ReadLine();

if (myName.Length == 0)

Console.WriteLine("Ви нічого не ввели, прощавайте");

else

Console.WriteLine("Здрастуйте, "+ myName);

Console.WriteLine("Скільки Вам років?");

int myAge = int.Parse(Console.ReadLine());

if (myAge == 0)

Console.WriteLine("Ви нічого не ввели, прощавайте");

else

{

if (myAge<15)

Console.WriteLine("Ви ще не студент");

else if (myAge>40)

Console.WriteLine("Вчитися ніколи не пізно!");

else

Console.WriteLine("Ваш вік" +myAge);

}

Console.ReadKey();

}

}

}

В цьому прикладі використовується декілька операторів розгалуження **if** (останні три вкладені). Зверніть увагу, що у C# в операторі **if** діють ті самі правила синтаксису, що і в С++, якщо в тілі оператора є лише один оператор блоку, то дужки не потрібні, наприклад:

if (myName.Length == 0)

Console.WriteLine("Ви нічого не ввели, прощавайте");

else

Console.WriteLine("Здрастуйте, "+ myName);

Крім цього виконується явне перетворення типів з типу **string** в тип **int** з використанням методу Parse і неявне при виведенні віку на консоль

#### 6. Створення консольних проектів з використанням операторів switch

Розробимо простий консольний калькулятор для виконання арифметичних операцій. Вхідні дані будемо вводити з консолі, результати виводити на консоль. Приклад коду з лекції

**Консольний калькулятор на 4 дії**

using System;

namespace ConsoleCalculator

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Hello World!");

string buf;

double a, b, res;

Console.WriteLine("Введіть перший операнд:");

a = double.Parse(Console.ReadLine());

Console.WriteLine("Введіть знак операції");

char op = (char)Console.Read();

Console.ReadLine();

Console.WriteLine("Введіть другий операнд:");

b = double.Parse(Console.ReadLine());

bool ok = true;

switch (op)

{

case '+': res = a + b; break;

case '-': res = a - b; break;

case '\*': res = a \* b; break;

case '/': res = a / b; break;

default: res = double.NaN; ok = false; break;

}

if (ok) Console.WriteLine("Результат: " + res);

else Console.WriteLine("Неприпустима операція");

}

}

}

Зверніть увагу, що тип виразу в операторі switch є **char**. Також зверніть увагу, що перетворення з типу string до типу char повинно бути явним, тому що тип char є типом-значенням, а тип string – посилковим.

**char op = (char)Console.Read();**

Для виходу з оператора switch при виконанні умови використовується оператор переходу break.

#### 7. Створення консольних проектів з використанням операторів циклу

**7.1. Цикл з передумовою while**

Розглянемо приклад з лекції 4, в якому використовується цикл з передумовою. Потрібно написати програму, яка виводить для аргументу х, що змінюється в заданих границях із заданим кроком, таблицю значень наступної функції:



Назвемо xn – початкове значення аргументу, xk –кінцеве значення аргументу, dx – крок зміни аргументу, t – параметр. Усі величини є дійсні числа типу double. Програма повинна виводити таблицю, що складається з двох стовпців: значень аргументу і відповідних ним значень функції.Таблиця повинна мати заголовок.

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

double Xn = -2, Xk = 12, dX = 2, t = 2, y;

Console.WriteLine("| x | y |"); // заголовок таблиці

double x = Xn;

while (x <= Xk)

{

y = t;

if (x >= 0 && x < 10) y = t \* x;

if (x >= 10) y = 2 \* t;

Console.WriteLine("| {0,6} | {1,6} |", x, y);

x += dX;

}

Console.ReadKey();

}

}

}

Зверніть увагу, що в тілі циклу використовується два оператори **if**. На кожному кроці циклу обчислюється значення функції. Цикл завершиться коли умова циклу не буде виконана (тобто значення x стане більше 12).

**3.2. Цикл з пост-умовою do…while**

Цей тип циклу застосовується в тих випадках, коли тіло циклу необхідно обов'язково виконати хоч б один раз.

*Приклад 4.7. з лекції 4* демонструє використання цього циклу. В цьому прикладі на консоль виводиться текст "Будете вчитися?" до тих пір поки не буду введено"y".

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

char answer;

do

{

Console.WriteLine("Будете вчитися?");

answer = (char)Console.Read();

Console.ReadLine();

} while (answer != 'y');

}

}

}

**3.3. Цикл з параметром (for)**

Цей тип циклу використовується коли відомі границі циклу (нижня і верхня). *Приклад 4.8 з лекції 3* демонструє використання цього циклу. В цьому прикладі обчислюється сума чисел від 1 до 100. Спочатку створюється змінна цілого типу s, яка ініціалізується нулем. Потім в тілі циклу на кожному кроці до неї додається чергове число: 1,2,3,....100.

int s = 0;

for ( int i = 1; i <= 100; i++ )

s += i;

Зверніть увагу, що в тілі циклу відсутні фігурні дужки блоку. Це тому, що в циклі лише один оператор. Також зверніть увагу на використання операції інкремента:

s += i;

замість призначення: s=s+i;

Обидва оператори дозволені в C# так само як і в C++.

Розглянемо ще один приклад використання оператора циклу з параметром.

**Постановка задачі.**

Написати метод обчислення функції sin(x), використовуючи розкладання в ряд Тейлора за формулою:

\sin x = \sum^{\infin}_{n=0} \frac{(-1)^n}{(2n+1)!} x^{2n+1}

При реалізації задачі використати цикл **for**.

Число x – це значення кута в радіанах, n – кількість членів ряду. Числа **x, n** ввести з консолі. Обчислення факторіала виконати в окремому методі. Порівняти отримане значення із стандартним методом обчислення sin(x).

В цьому прикладі для обчислення ступеня використовується метод Math.Pow(). Статичний клас Math містить методи реалізації математичних функцій.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace My\_sin

{

class Program

{

static double Calc\_sin(double x, int n)

{

//обчислення розкладання sin в ряд

double result =0;

for (int i = 0; i < n; i++)

{

result=result+(Math.Pow((-1),i)\*Math.Pow(x,(2\*i+1)))/F(2\*i+1);

}

return result;

}

static double F(int n)

{

double tmp = 1;

for (int i = 1; i <= n; i++)

{

tmp = tmp \* i;

}

return tmp;

}

static void Main(string[] args)

{

Console.WriteLine("Введіть x - кут в радіанах");

double x = double.Parse(Console.ReadLine());

Console.WriteLine("Введіть показник ступеня n");

int n = int.Parse(Console.ReadLine());

//виклик методу обчислення sin(x) через ряд

double my\_sinus = Calc\_sin(x,n);

//виклик методу з класу Math

double sinus = Math.Sin(x);

double delta = sinus - my\_sinus;

Console.WriteLine("my\_sinus= {0},sin={1},delta={2}", my\_sinus, sinus, delta);

Console.ReadKey();

}

}

}

### Варіанти завдань для лабораторної роботи № 1.

**Номер варіанту відповідає номеру прізвища студента у списку групи**

Написати в C# консольний застосунок, що реалізує завдання згідно з варіантом.

Виконання завдання 1 вимагає використання методів класу **Console.**

Виконання завдання 2 вимагає використання методів перетворення типів.

Виконання завдання 3 вимагає використання методів класу **Math.**

Виконання завдання 4 вимагає використання оператору **if**

Виконання завдання 5 вимагає використання оператору **switch**

Виконання завдання 6 вимагає використання оператору циклу

Виконання цієї роботи може зайняти 30 хвилин вашого часу.

Кожний студент повинен виконати по 5 завдань.

|  |  |
| --- | --- |
| **№ варіанту** | **Зміст завдання** |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Обчислення значення поліному *p = a\*x^5 − 1 / b\*x^4+c\*x + d*. Значення *a, b, c, d* – дійсні числа, які слід ініціалізувати у функції, значення змінної *x* увести з консолі. Результат вивести на консоль. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. Обчислити значення функції в точці *х*, значення якої ввести з консолі, Якщо введено не число, вивести повідомлення «Ввести число».      1. Написати функцію, яка в залежності від порядкового номера місяця (1,2,...12) виводить на екран його назву (січень,...грудень). 2. Дано натуральне число n. Обчислити добуток перших n членів ряду. |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Обчислення значення поліному *p = a\*x^4 − b\*x^3 + c\*x + d*. Значення *a, b, c, d* – дійсні числа, які слід ініціалізувати у функції, значення змінної *x* увести з консолі. Результат вивести на консоль. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. Обчислити значення функції в точці *х*, значення якої ввести з консолі:      1. Написати функцію, яка в залежності від порядкового номера дня тижня (1,2,...7) виводить на екран його назву (понеділок,...) 2. Обчислити суму чисел в заданому у функції діапазоні. Значення чисел вводити з консолі. Якщо значення, що вводиться, виходить за межі заданого діапазону, вивести повідомлення про помилку і повторити введення. Не застосовувати масиви. |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Обчислення значення поліному p=23,5\*x^5 + 30\*x^4+10\*x+87,3. Значення *a, b, c, d* – дійсні числа, які слід ініціалізувати у функції, значення змінної *x* увести з консолі. Результат вивести на консоль. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. Обчислити значення функції в точці *х*, значення якої ввести з консолі:      1. Написати функцію, яка в залежності від порядкового номера місяця (1,2,...12) виводить на екран пору року (зима, весна....) 2. Дано натуральне число n. Обчислити   , |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Обчислення значення поліному p = 3,5\*x^4 + 3\*x^3+10\*x^2+8,3. Значення *a, b, c, d* – дійсні числа, які слід ініціалізувати у функції, значення змінної *x* увести з консолі. Результат вивести на консоль. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. Обчислити значення функції в точці *х*, значення якої ввести з консолі:      1. Написати функцію, яка в залежності від одиниці виміру часу (1,2,...24) виводить на екран частини доби (ранок, день,....) 2. Задати з консолі оцінки студента з 10 дисциплін. Обчислити загальну суму балів, найгіршу і найкращу оцінку. Не використовувати масиви. |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Обчислити площу прямокутника за заданими сторонами. Результат вивести на консоль. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. За даними *a, b, x,* значення яких ввести з консолі, обчислити значення функції:      1. Написати функцію, яка в залежності від порядкового номера кольору у спектрі (1,2,...7) виводить його назву (червоний, помаранчевий, жовтий, зелений, блакитний, синій, фіолетовий) і код RGB. 2. N суддів поставили різні оцінки одному спортсмену. Обчислити середній бал спортсмена, видаливши найменшу та найвищу суддівські оцінки. Оцінки вводити з клавіатури, Не використовуючи масиви, обраховувати суму введених значень, найменше та найбільші значення. |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Знайти довжини всіх медіан і бісектрис трикутника, якщо значення сторін трикутника введені з клавіатури. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. Обчислити значення функції в точці *х*, значення якої ввести з консолі:      1. Написати функцію, яка виводить назву навчальної дисципліни в залежності від уподобань студента, що задаються пріоритетом 1,2,...5 дисципліни. 2. Дано натуральне число n і дійсне число x >0. Обчислити суму для заданого х: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Обчислити відстань від точки (x0, y0) до точки (х1, y1), значення координат яких введені з консолі. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. Обчислити значення функції в точці *х*, значення якої ввести з консолі:      1. Написати функцію, яка виводить на консоль назву навчального закладу в залежності від середнього балу ЗНО та пріоритетів (1,2,…,5), заданих абітурієнтом. 2. Обчислити  - число Фібоначчі з номером *n*, де: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Знайти об'єм циліндра, якщо значення радіусу його основи та висоти введені з консолі. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. Обчислити значення функції в точці *х*, значення якої ввести з консолі:      1. Написати функцію, яка виводить на консоль назву університету в залежності від його консолідованим рейтингом (1,2,…,5) за 2020, що визначається інформаційним ресурсом "Освіта.ua". 2. Дано натуральне число n. Обчислити суму перших 2n членів ряду: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Обчислити відстань від точки (x0, y0) до прямої *ax + by + c*= 0. Значення координат точки та коефіцієнтів a, b, c прямої ввести з консолі. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. За даними *a, b, x,* значення яких ввести з консолі, обчислити значення функції:      1. Написати функцію, яка виводить на консоль назву країни в залежності від рейтингу (1,2,…,5) щастя її населення за 2020, що визначається Social Progress Index (<https://minfin.com.ua/ua/2020/10/11/53857422/> ) 2. Дано натуральне число n і дійсне число x >0. Обчислити суму перших 2n членів ряду: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Знайти довжини всіх висот трикутника, якщо значення сторін a, b, c трикутника введені з консолі 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. За даними *a, b, x,* значення яких ввести з консолі, обчислити значення функції:      1. Написати функцію, яка виводить на консоль прізвище студента за його семестровим рейтингом (від 60 до 100) за результатами сесії. 2. Дано натуральне число n і дійсне число x >0. Обчислити суму членів ряду: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Знайти об'єм конуса, якщо значення радіусу його основи та висоти ведені з консолі. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. За даними *a, b, x,* значення яких ввести з консолі, обчислити значення функції:      1. Написати функцію, яка в залежності від назви кольору та сигналів світлофора виводить на консоль назву дії, яку має виконати водій автомобіля (їхати, чекати, зупинитися, повертати, зменшити швидкість тощо). 2. Дано натуральне число n і дійсне число x >0. Обчислити суму членів ряду: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Обчислити площу поверхні () сфери за значенням з консолі радіусу **r**. Результат вивести на консоль. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. За даними *a, b, x,* значення яких ввести з консолі, обчислити значення функції:      1. Написати функцію, яка в залежності від номеру ІТ-спеціальності (121, 121, …, 126) виводить на консоль її назву. 2. Дано натуральне число n і дійсне число x >0. Обчислити добуток членів ряду: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Тіло починає рухатися без початкової швидкості з прискоренням *a*. Обчислити відстань, яку тіло пройде за час *t* від початку руху. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. За даними *a, b, с, x,* значення яких ввести з консолі, обчислити значення функції:      1. Написати функцію, яка в залежності від назви місяця (січень, лютий, …) виводить на консоль його порядковий номер. 2. Дано натуральне число n і дійсне число x >0. Обчислити суму членів ряду: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Висота ромба, проведена з вершини тупого кута, ділить сторону навпіл. Знайдіть меншу діагональ, якщо значення периметра ромба введене з клавіатури. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. За даними *a, x,* значення яких ввести з консолі, обчислити значення функції:      1. Написати функцію, яка в залежності від назви кольору у спектрі (червоний, помаранчевий, жовтий, зелений, блакитний, синій, фіолетовий) виводить його порядковий номер і код RGB. 2. Дано натуральне число n і дійсне число x >0. Обчислити суму членів ряду: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати **в одному проекті Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Написати функцію, яка визначає, чи пройде куля радіуса *r* в квадратний отвір зі стороною *a*. Дійсні значення r, a увести з консолі. Вивести на консоль відповідне повідомлення. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. За даними *x,* значення яких ввести з консолі, обчислити значення функції:      1. Написати функцію, яка в залежності від номеру року виводить на консоль назву країни, співак якої став переможцем конкурсу Євробачення, наприклад, у 2016 році перемогла Україна, 2017 рік - Португалія, і т. д. 2. Дано натуральне число n і дійсне число x >0. Обчислити суму членів ряду: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою **меню**, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати в одному проекті **Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Написати функцію, яка визначає, чи пройде куб з ребром *a* в круглий отвір радіуса *r*. Дійсні значення *а,* *r* увести з консолі. Вивести на консоль відповідне повідомлення. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. За даними x, значення яких ввести з консолі, обчислити значення функції:      1. Написати функцію, яка в залежності від назви телеканалу виводить на консоль назву холдингу (власника), наприклад, канал «Рада», власник «Верховна Рада України», канал «1+1», холдинг «1+1 Media» і т. д. 2. Дано натуральне число *n* і дійсне число *x* >0. Обчислити суму членів ряду: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою **меню**, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати в одному проекті **Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Написати функцію, яка визначає, з якою швидкістю спортсмен увійде у воду, стрибаючи з *n* метрової вежі, якщо спортсмен падає з прискоренням a=9,81м/с^2, початкова швидкість v0 = 0? Значення *n* увести з консолі. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. За даними x, значення яких ввести з консолі, обчислити значення функції:      1. Написати функцію, яка в залежності від номеру пальця на руці виводить його назву (великий, вказівний, середній, безіменний, мізинець). 2. Дано натуральне число *n* і дійсне число *x* >0. Обчислити суму членів ряду: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою **меню**, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати в одному проекті **Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Написати функцію, яка визначає, подібність двох трикутників за трьома сторонами, значення яких введені з консолі. Вивести на консоль відповідне повідомлення. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. За даними x, значення яких ввести з консолі, обчислити значення функції:      1. Написати функцію, яка в залежності від назви країни виводить її рейтинг та індекс рівня освіти (<https://gtmarket.ru/ratings/education-index> ). 2. Дано натуральне число *n* і дійсне число *x* >0. Обчислити суму членів ряду: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою **меню**, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати в одному проекті **Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Написати функцію, яка визначає, подібність двох трикутників за двома сторонами та кутом між ними. Значення двох сторін та кутів трикутників увести з консолі. Вивести на консоль відповідне повідомлення. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. За даними x, значення яких ввести з консолі, обчислити значення функції:      1. Написати функцію, яка в залежності від назви країни виводить її рейтинг шкільної освітньої грамотності за версією PISA (<https://factsmaps.com/pisa-2018-worldwide-ranking-average-score-of-mathematics-science-reading/> ). 2. Дано натуральне число *n* і дійсне число *x* >0. Обчислити суму членів ряду: |
|  | Створити консольний застосунок мовою C#. Вхідні дані ввести з клавіатури. Результати вивести на консоль Використати методи класів **Console, Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою **меню**, застосувавши оператор вибору **switch** для виклику потрібної функції. Усі завдання варіанта реалізувати в одному проекті **Console\_Lab1**.   1. Вивести на консоль власні анкетні дані: прізвище, ім'я, вік, група, курс, e=mail. 2. Написати функцію, яка визначає приналежність точки з координатами (*x,y*) колу з радіусом *r* та центром на початку координат. Значення x,y,r ввести з консолі. 3. За даними, що введені з консолі, визначити значення виразу, використовуючи математичні функції, і вивести результат на консоль.      1. За даними x, значення яких ввести з консолі, обчислити значення функції:      1. Написати функцію, яка в залежності від назви країни виводить її рейтинг освітньої грамотності за версією PISA (). 2. Дано натуральне число *n* і дійсне число *x* >0. Обчислити суму членів ряду: |

### Література

1. О.С.Бичков, Є.В.Іванов Об’єктно-орієнтоване програмування мовою C#/

2.C# 2005 и платформа .Net 3.0 для профессионалов. Нейгел К., Ивьен Б. и др. –М.: ООО “И.Д. Вильямс", 2008. –1376 с.

## Лабораторна робота № 2 Мова C#. Масиви, матриці, рядки. Методи класів Console, Convert, Random, Math

### Мета роботи:

1. Робота з масивами в C#

2. Оператор foreach

3. Генерація випадкових чисел

4. Робота з математичними функціями

5. Лінійний пошук

6. Двійковий пошук

### Методичні рекомендації до виконання лабораторної роботи

1. Прочитайте лекцію.

2. Прочитайте цю лабораторну роботу та виконайте наведені в ній приклади (вони всі працездатні)

3. При виконанні завдань зверніть увагу на приклад застосування до масивів циклу **foreach**

4. Також зверніть увагу на опис і реалізацію методу бісекцій (ділення навпіл) знаходження коренів нелінійних рівнянь. Те також задача на одновимірні масиви і цикли

5. Для поглибленого вивчення цього матеріалу прочитайте [1, 2].

### Порядок виконання роботи

1. Створити директорію Lab2, в якій будуть розміщуватися проекти цієї лабораторної роботи.

2. Виконати завдання свого варіанту у вигляді одного консольного проекту

3. Зафіксувати результати у Контрольній роботі 1

### Приклади виконання завдань

#### 1. Робота з масивами в C#

Розглянемо приклад програми з лекції, яка визначає суму і кількість від'ємних елементів, а також максимальний елемент масиву, що складається з 6 цілочисельних елементів.

Тут елементи масиву **а** ініціалізуються при створенні масиву. Далі в циклі з параметром елементи масиву виводяться на консоль. Зверніть увагу, що для виводу використовується метод Console.Write а не Console.WriteLine, тому всі елементи виводяться в один рядок. Символ табуляциії "\t" в методі Console.Write() розділяє елементи при виводі.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_Lab4

{

class Program

{

static void Main(string[] args)

{

const int n = 6;

int[] a = new int[n] { 3, 12, 5, -9, 8, -4 };

Console.WriteLine( "Початковий масив:" );

for ( int i = 0; i < n; ++i )

Console.Write( "\t" + a[i] );

Console.WriteLine();

long sum = 0; // сума від'ємних елементів

int num = 0; // кількість від'ємних елементів

for ( int i = 0; i < n; ++i )

if ( a[i] < 0 )

{

sum += a[i];

++num;

}

Console.WriteLine( "Сума від'ємних = " + sum );

Console.WriteLine( "Кількість від'ємних = " + num );

int max = a[0]; // максимальний елемент

for ( int i = 1; i < n; ++i )

if ( a[i] > max ) max = a[i];

Console.WriteLine( "Максимальний елемент = " + max );

Console.ReadKey();

}

}

}

Розглянемо ще одну "класичну" задачу – генерація чисел Фібоначчі: послідовності чисел, яка задовольняє умовам

F1 = 1; F2 = 1; Fk = Fk-1 + Fk-2 для k>2.

Цю задачу можна реалізувати з використанням масивів. В наступному прикладі кількість чисел вводиться з консолі, потім створюється одновимірний масив вказаної розмірності, який заповнюється згенерованими числами.

##### Генерація чисел Фібоначчі

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_Lab4

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введіть кількість чисел послідовності");

int n = int.Parse(Console.ReadLine());

int[] fibonachi = new int[n];

fibonachi[0] = 1;

fibonachi[1] = 1;

for (int i = 2; i < n; i++)

{

fibonachi[i] = fibonachi[i - 2] + fibonachi[i - 1];

}

for (int i = 0; i < n; i++)

Console.WriteLine("fibonachi[" + i + "]=" + fibonachi[i]);

Console.ReadKey();

}

}

}

##### Генерація простих чисел. Алгоритм Ератосфена

Простим є число, яке ділиться тільки на 1 і на себе. Алгоритм пошуку послідовності простих чисел запропонував древньогрецький математик Ератосфен, і він отримав назву **Решето Ератосфена.**

**Ідея і загальний опис алгоритму.**

Є розташована в ряд за збільшенням послідовність цілих чисел. Спочатку в ній викреслюються усі числа кратні 2, окрім її самої, і так до N. Далі із списку, що вийшов, береться число, що йде за двійкою, - трійка, викреслюються усі кратні 3 числа, окрім її самої. У такому вигляді алгоритм триває для частини послідовності, що залишилася, і у результаті отримаємо усі прості числа у вказаному діапазоні.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 22 | 33 | 44 | 55 | 66 | 77 | 88 | 99 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 220 |
| 221 | 222 | 223 | 224 | 225 | 226 | 227 | 228 | 229 | 330 | 331 | 332 | 333 | 334 | 335 | 336 | 337 | 338 | 339 | 440 |
| 441 | 442 | 443 | 444 | 445 | 446 | 447 | 448 | 449 | 550 | 551 | 552 | 553 | 554 | 555 | 556 | 557 | 558 | 559 | 660 |
| 661 | 662 | 663 | 664 | 665 | 666 | 667 | 668 | 669 | 770 | 771 | 772 | 773 | 774 | 775 | 776 | 777 | 778 | 779 | 880 |
| 881 | 882 | 883 | 884 | 885 | 886 | 887 | 888 | 889 | 990 | 991 | 992 | 993 | 994 | 995 | 996 | 997 | 998 | 999 | 1100 |

У таблиці наведені усі цілі числа від 2 до 100. Червоним помічені ті, які були видалені в процесі виконання алгоритму Решето Ератосфену.

Тепер розглянемо алгоритм детальніше, розбивши його на декілька частин. Отже, для пошуку простих чисел методом Решета Ератосфену треба:

1. Організувати список з чисел від 2 до N, а також логічний масив розмірністю N;
2. У вільну змінну R записати число 2;
3. Виключити усі числа кратні R, починаючи з R\*2;
4. Записати в R наступне за R не закреслене число;

Повторювати дії, описані в двох попередніх кроках.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace SimpleNumber

{

class Program

{

//Нахождение простых чисел. Алгоритм "Решето Эратосфена"

//Запишем подряд все числа от 2 до N.

//Дальше вычеркнем из этого списка все числа кратные 2,

// исключая саму двойку, потом вычеркнем все числа кратные 3,

//исключая само число 3, число 4 уже вычеркнуто, вычеркиваем числа

// кратные 5 и т.д.

//Продолжаем этот процесс, пока квадрат очередного числа не превысит N.

static void Main(string[] args)

{

bool[] table = new bool[100];

int i, j;

// Отмечаем все числа как простые

for (i = 0; i < table.Length; i++)

table[i] = true;

// Вычеркиваем лишнее

for (i = 2; i \* i < table.Length; i++)

if (table[i])

for (j = 2 \* i; j < table.Length; j += i)

table[j] = false;

// Выводим найденное

for (i = 2; i < table.Length; i++)

{

if (table[i])

Console.WriteLine(i);

}

Console.ReadKey();

}

}

}

#### 2. Оператор foreach

Оператор **foreach** використовується для перебору елементів в масивах та інших колекціях і є новим типом циклу, реалізованому в C#. В цьому прикладі створюється *ступінчастий масив*, який заповнюється даними. За допомогою вкладених операторів циклу елементи масиву виводяться на консоль у вигляді матриці.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_Lab4

{

class Program

{

static void Main(string[] args)

{

//оператор foreach і ступінчасті масиви

int[][] а = new int[3][];

а[0] = new int[5] { 24, 50, 18, 3, 16 };

а[1] = new int[3] { 7, 9, -1 };

а[2] = new int[4] { 6, 15, 3, 1 };

Console.WriteLine("Початковий масив:");

foreach (int[] mas1 in а)

{

foreach (int x in mas1)

Console.Write("\t" + x);

Console.WriteLine();

}

Console.ReadKey();

}

}

}

#### 3. Генерація випадкових чисел

Потреба в генерації випадкових чисел часто виникає в програмуванні. В C# є *нестатичний* клас Random, методи якого дозволяють генерувати різні послідовності випадкових чисел. Нижче наведено приклад з лекції 4 в якому в методі ValsGenerator створюється об'єкт **aRand**. Далі в циклі викликається метод **aRand.Next(100),** який при кожному виклику генерує одне випадкове число від 1 до100 і призначає його черговому елементу масиву. Зверніть увагу, що метод **ValsGenerator** є статичним.

В методі **Main** створюється масив **Data,** викликається метод **ValsGenerator** для його заповнення, потім викликається метод Array.Sort(Data) для його сортування. Після чого елементи масиву виводяться в циклі на консоль.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Console\_Lab4

{

class Program

{

// генератор даних

static void ValsGenerator(int[] Vals)

{

// Random - клас для генерації випадкових чисел

Random aRand = new Random();

// заповнення масиву

for (int i = 0; i < Vals.Length; i++)

Vals[i] = aRand.Next(100);

}

static void Main(string[] args)

{

const int N = 10;

int[] Data = new int[N];

ValsGenerator(Data);

Array.Sort(Data);

Console.WriteLine("Друк відсортованих даних");

for (int i = 0; i < Data.Length; i++)

Console.WriteLine("Data[" + i + "] = " + Data[i]);

Console.ReadLine();

}

}

}

Зверніть увагу, що при виводі масиву

Console.WriteLine("Data[" + i + "] = " + Data[i]);

виконується неявне приведення з типу int до типу string.

#### 4. Робота з математичними функціями

Клас Math містить методи для роботи з математичними функціями.

Розглянемо ще один приклад – використання масивів і математичних функцій в задачах обчислювальної математики.

**Постановка задачі**

Знайти дійсні корені рівняння 6x4-3x3+8x2-5=0 за методом бісекції (ділення навпіл) на відрізках [0, 1], [-1, 0].

**Алгоритм методу:**

Нехай [а,b] відрізок, на якому шукають корені. Припустимо, що функція f(x) неперервна на [а,b] і на кінцях приймає значення різних знаків *Алгоритм* методу полягає в побудові послідовності вкладених відрізків, на кінцях яких функція приймає значення різних знаків. Кожний наступний відрізок отримують діленням навпіл попереднього. Опишемо один крок ітераційного методу. Нехай на *к-*ому кроці знайдено відрізок такий, що . Знайдемо середину відрізку . Якщо, то - корінь і задача вирішена. Якщо ні, то з двох половин відрізку вибираємо той, на кінцях якого функція має протилежні знаки:



, , якщо



, , якщо



*Критерій закінчення ітераційного процесу*: якщо довжина відрізку знаходження кореня менше 2, то ітерації припиняють і за значення кореня із заданою точністю приймають середину відрізку.



Код програми може бути таким:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace bicection

{

class Program

{ //Знайти корені нелінійного рівняння

//6x4-3x3+8x2-5=0

//x1=0.74213

//x2=-0.6365

static double f(double x)

{

//рівняння, для якого шукаємо корені

double y=6\*Math.Pow(x,4)-3\*Math.Pow(x, 3)+8\*Math.Pow(x,2)-5;

return y;

}

// --------------------------------------------------------------------

static double bicect(double left, double right)

{

//метод бісекцій

double eps = 0.00001;

double center = 0;

while (right - left > eps \* 2)

{

center = (right - left) / 2 + left;

if (f(center) \* f(left) > 0)

left = center;

else

right = center;

}

return center;

}

// ----------------------------------------------------

static void Main(string[] args)

{

//метод бісекцій

double x1 = bicect(0, 1);

double x2 = bicect(-1, 0);

Console.WriteLine("Метод бісекцій");

Console.WriteLine("x1={0},x2= {1}", x1, x2);

Console.ReadKey();

}

}

}

В цій програмі є два методи: static double f(double x), в якому визначається рівняння, і метод static double bicect(double left, double right). Для обчислення ступеня в методі **f** використовується метод Math.Pow( ).

Другий метод static double bicect(double left, double right) реалізує ітераційний процес. В методі Main два рази викликається метод бісекцій для знаходження коренів на вказаних відрізках. Результати виводляться на консоль.

#### 5. Лінійний пошук в масиві

**Лінійний, послідовний пошук** — алгоритм знаходження заданого значення довільної функції на деякому відрізку.

**Формальний запис алгоритму:**

1. Визначити елемент, який шукаємо ***key;***

2. Встановити границі області пошуку L, R для масиву A [a];

3. Якщо аi = key – елемент знайдено;

4. Інакше – переходимо до наступного елементу.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace Search

{

class Program

{

static void Main(string[] args)

{

int i, n, key, nom;

bool x = false;

Console.WriteLine("Розмір масиву");

n = int.Parse(Console.ReadLine());

int[] mas = new int[n]; //визначення масиву

Console.WriteLine("Шуканий елемент");

key = int.Parse(Console.ReadLine());

// Random - клас для генерації випадкових чисел

Random aRand = new Random();

for (i = 0; i < n; i++)

{

//формування масиву, заповнення його випадковими числами

mas[i] = aRand.Next(n);

Console.Write(mas[i] + " ");

}

for (i = 0; i < n; i++)

{ //якщо цей елемент дорівнює шуканому

if (mas[i] == key) //то x призначаємо true

{ x = true; nom = i; break; } //і виходимо з циклу

}

if (x == true)

Console.WriteLine("Елемент знайдено");

else Console.WriteLine("\nЕлемент не знайдено");

Console.ReadKey();

}

}

}

#### 6. Двійковий (бінарний) пошук

**Двійковий (бінарний) пошук** (також відомий як метод ділення навпіл і дихотомія) — класичний алгоритм пошуку елемента у відсортованому масиві.

**Формальний запис алгоритму:**

1. Масив ділитися на дві рівні частини, шляхом визначення першого (a), останнього (b) і середнього (c) елементів;
2. Середній елемент порівнюється з шуканим (s):

якщо **s<c.** Останньому елементу призначається значення середнього, тим самим ділянка пошуку зменшується удвічі: b=c;

якщо **s>c**. Першому елементу призначається значення середнього, і ділянка пошуку зменшується удвічі: a=c;

якщо **s=c**. Елемент знайдений, і робота алгоритму завершується**.**

1. Якщо для перевірки не залишився жодного елементу, то алгоритм завершується, інакше виконується перехід до пункту 2.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace BinarySearch

{

class Program

{

static void Main(string[] args)

{

int i, n, key, begin, end, c;

bool x = false;

Console.WriteLine("Розмір масиву");

n = int.Parse(Console.ReadLine());

int[] mas = new int[n]; //визначення масиву

Console.WriteLine("Шуканий елемент");

key = int.Parse(Console.ReadLine());

for (i = 0; i < n; i++)

{

//формування масиву, заповнення його числами кратними n (10,20,....)

mas[i] = n \* i;

Console.Write(mas[i] + " ");

}

begin = 0; end = n; //ліва і права границі масиву

while (begin < end)

{

// власне реалізація алгоритму

c = begin + (end - begin) / 2;

if (key < mas[c]) end = c;

else if (key > mas[c]) begin = c + 1;

else { x = true; break; }

}

if (x == true)

Console.WriteLine("Елемент знайдено");

else Console.WriteLine("\nЕлемент не знайдено");

Console.ReadKey();

}

}

}

### Варіанти завдань для лабораторної роботи № 2

Завдання передбачає реалізацію 10 функцій. Функції 1 – 5 передбачають роботу з одновимірними масивами, функції 6 – 9 передбачають опрацювання матриць, функція 10 – робота з рядком символів типу string.

|  |  |
| --- | --- |
| **№ варі-анту** | **Зміст завдання** |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість з консолі. Відсортувати згенерований масив за спаданням значень його елементів алгоритмом швидкого сортування. Вивести на консоль масив до та після сортування. 2. У згенерованому масиві визначити прості числа із заданого з консолі діапазону, використавши алгоритм Ератосфена, та вивести їх на консоль. 3. Визначити кількість повторень кожного елементу масиву, застосувавши метод лінійного пошуку. 4. Вивести на консоль значення мінімального та максимального елементів масиву та їх індекси, застосувавши метод лінійного пошуку. 5. Вивести на консоль значення елемента, заданого з консолі, застосувавши метод бінарного пошуку. У разі його відсутності в масиві ви вести відповідне повідомлення. Модифікувати функцію двійкового пошуку у масиві, використавши метод BinarySearch класу Array. 6. Згенерувати елементи матриці, задавши її вимірність з консолі. Вивести матрицю на консоль у вигляді таблиці. Визначити суму елементів заданого з консолі номеру рядка матриці та суму елементів заданого з консолі номеру стовпчика матриці. Вивести на консоль номери рядка та стовпчика матриці та обчислені суми. 7. Вивести на консоль значення мінімального та максимального елементів матриці та їх індекси, застосувавши методи класу Math. 8. Знайти корені нелінійного рівняння 6x4−3x3+8x2−25=0, застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів (тип string). Виконати операції редагування рядка, здійснивши пошук та заміну підрядка на заданий з консолі підрядок, вставку підрядка, видалення заданого підрядка. Вивести на консоль рядок після редагування. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом бульбашкового сортування. Вивести масив до та після сортування на консоль. 2. Побудувати новий одновимірний масив простих чисел, використавши алгоритм Ератосфена, та вивести їх на консоль. 3. Об’єднати згенерований раніше масив Обчислити суму елементів масиву, середнє арифметичне елементів, min і max елементів масиву, використавши методі класу Math. Результати вивести на консоль. 4. Вивести на консоль індекси усіх повторень шуканого елемента масиву, застосувавши метод лінійного пошуку. Значення шуканого елемента ввести з консолі 5. Вивести на консоль індекс елемента, значення якого задане з консолі, застосувавши метод бінарного пошуку. У разі його відсутності в масиві ви вести відповідне повідомлення. Модифікувати функцію двійкового пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати матрицю, i–й рядок якої визначає номер співробітника, j-й стовпчик якої визначає номер місяця року. Кількість співробітників задати з консолі. Значення на перетині i-го рядка та j-го стовпця визначає зарплату i-го співробітника у j-му місяці. Вивести матрицю на консоль у вигляді таблиці. Визначити загальний бюджет зарплати за рік, загальну і середню зарплату за місяць, номер якого ввести з консолі. 7. У згенерованій матриці поміняти місцями рядки, що містять максимальний та мінімальний елементи. Якщо шукані елементи знаходяться в одному рядку, то поміняти місцями стовпчики. Вивести на консоль матрицю до та після переміни місцями рядків чи стовпчиків. 8. Знайти корені нелінійного рівняння (x^2−6\*x)^2−2(x−3)^2=81, застосувавши метод Ньютона. Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів (тип string). Виконати операції редагування рядка, здійснивши заміну усіх алфавітних символів на їх ASCII коди та видаливши усі цифрові символи. Вивести рядок після заміни на консоль. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом сортування вибором. Вивести на консоль масив до та після сортування. 2. Вивести значення індексів елементів масиву, що не змінилися після його сортування, та їх кількість. 3. Створити новий масив натуральних чисел, значеннями якого є прості числа в діапазоні від 1 до N. Значення N задати з консолі. Використати алгоритм Ератосфена для визначення простих чисел. Вивести прості числа на консоль. Якщо в згенерованому раніше масиві відсутні знайдені за алгоритмом Ератосфена прості числа, то вставити їх в згенерований масив, не змінивши його упорядкованість за зростанням. Вивести отриманий масив на консоль. 4. У сформованому масиві визначити кількість парних елементів і елементів з парними індексами, застосувавши алгоритм лінійного пошуку. 5. Визначити кількість і значення елементів масиву, що кратні заданому з консолі числу, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Модифікувати функцію двійкового пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень з консолі. На перетині i-го рядка та j-го стовчика матриці записаний прибуток за j-й місяць від i-го проекту, який виконується в компанії. Вивести матрицю на консоль. Визначити загальний прибуток від кожного проекту, загальний дохід компанії від усіх проектів за усі місяці, індекс проекту з найбільшим прибутком. Вивести результати на консоль. 7. У створеній матриці визначити індекси та значення її сідлових точок. Елемент матриці називається сідловою точкою, якщо він є найменшим у своєму рядку та найбільшим у своєму стовпчику. Вивести на консоль значення та індекси її сідлових точок або повідомлення про відсутність таких елементів. 8. Знайти корені нелінійного рівняння (x^2+x+1)(x^2+x+1)−12=0, застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів (тип string). Порахувати кількість слів, що починаються з великої літери. Вивести текст, складений з останніх букв усіх слів. Результати вивести на консоль. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math**   1. Згенерувати додатні та від’ємні значення елементів одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом сортування включенням. Вивести на консоль масив до та після сортування. 2. Визначити прості числа серед додатних елементів масиву, використавши алгоритм Ератосфена. Вивести прості числа на консоль. Якщо в згенерованому масиві відсутні прості числа, то вивести на консоль відповідне повідомлення. 3. Переставити елементи масиву так, щоб спочатку були розташовані всі від’ємні елементи, потім усі додатні елементи, потім усі нульові. Порядок серед додатних і від’ємних елементів має зберегтися. Надрукувати масив після переставлення елементів. 4. Знайти найбільший серед від’ємних та найменший серед додатних елементів масиву, застосувавши алгоритм лінійного пошуку. Вивести значення знайдених елементів та їх індекси. 5. Визначити кількість і значення елементів масиву, що кратні заданому з консолі числу, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень з консолі. На перетині i-го рядка та j-го стовчика матриці записаний прибуток за j-й місяць від продажу i-го товару в магазині. Вивести матрицю на консоль. Визначити загальний прибуток від кожного товару, загальний прибуток магазину від продажу усіх товарів за усі місяці, індекс товару, який приносить прибуток. Вивести результати на консоль. 7. Вивести на консоль значення мінімального та максимального елементів матриці та їх індекси, застосувавши методи класу Math. 8. Знайти корені нелінійного рівняння (x^2−5\*x+7)^2−(x−2)(x−3)=0, застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів (тип string), що містить круглі, квадратні та фігурні дужки. Визначити чи є послідовність дужок правильною, тобто кількість дужок, що відкривається, дорівнює кількості дужок, що закриваються. Результати вивести на консоль. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math**   1. Згенерувати цілочислові додатні та від’ємні елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом сортування Шелла. Вивести на консоль масив до та після сортування. 2. У згенерованому масиві визначити прості числа із заданого з консолі діапазону додатних цілих значень, використавши алгоритм Ератосфена, та вивести їх на консоль. Якщо в згенерованому масиві відсутні прості числа, то вивести на консоль відповідне повідомлення 3. Переставити елементи масиву так, щоб їх значення чергувалися в порядку: від’ємний, додатний, нульовий, від’ємний, додатний, нульовий і т.д, Надрукувати масив після переставлення елементів. 4. Знайти найбільший серед від’ємних та найменший серед додатних елементів масиву, застосувавши алгоритм лінійного пошуку. Вивести значення знайдених елементів та їх індекси. 5. Визначити кількість і значення елементів масиву, що належать заданому з консолі діапазону, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень з консолі. На перетині i-го рядка та j-го стовпчика матриці записаний прибуток за j-й місяць від продажу i-го товару в магазині. Вивести матрицю на консоль. Визначити загальний прибуток від кожного товару, загальний прибуток магазину від продажу усіх товарів за усі місяці, індекс товару, який приносить прибуток. Вивести результати на консоль. 7. У створеній матриці визначити та вивести на консоль значення та індекси мінімального елемента. Видалити рядок та стовпчик з мінімальним елементом матриці. Вивести на консоль отриману після перетворення матрицю. 8. Знайти корені нелінійного рівняння (x^2−5\*x+7)^2−(x−2)(x−3)=0, застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів (тип string), що містить круглі, квадратні та фігурні дужки. Визначити, чи є послідовність дужок правильною, тобто кількість дужок, що відкривається, дорівнює кількості дужок, що закриваються. Результати вивести на консоль. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати методи класу **Math**   1. Згенерувати натуральні значення елементів одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом бульбашкового сортування. Вивести на консоль масив до та після сортування. 2. Перевірити, чи елементи згенерованого масиву, які мають непарні індекси, є простими числами. Для визначення простих чисел використати алгоритм Ератосфена, Якщо такі числа знайдені, вивести їх на консоль, інакше вивести відповідне повідомлення. 3. Переставити елементи масиву так, щоб парні і непарні елементи чергувалися. Перед кожним простим числом вставити нульове значення. Вивести на консоль масив після переставлення елементів. 4. Знайти у відсортованому масиві елементи, значення яких є трикутними числами, що визначаються за формулою , де n – порядковий номер числа в послідовності трикутних чисел (1, 3, 6, 10,…). Застосувати алгоритм лінійного пошуку. Результати вивести на консоль. 5. Визначити кількість і значення елементів масиву, що належать заданому з консолі діапазону, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів з консолі, включаючи нульове значення. На перетині i-го рядка та j-го стовпчика матриці записана кількість вакцинованих людей проти Covid-19 в *i*-й країні в *j*-му місяці. Якщо значення дорівнює нулю, це означає, що в *i*-й країні в *j*-му місяці людей не вакцинували. Вивести матрицю на консоль. Визначити загальну кількість людей, що вакциновані в усіх країнах за весь період, країну, в якій найменша кількість вакцинованих за весь період, місяць з найбільшою кількістю вакцинованих. Результати вивести на консоль. 7. Із зенерованої матриці видалити країні, які за весь період не вакцинували жодної людини, переставити стовпці матриці відповідно до зростання сум їх елементів. Перетворену матрицю вивести на консоль. 8. Знайти корені нелінійного рівняння 2x^4−x^3−9x^2+13x−5=0, застосувавши метод бісекції. Результати вивести на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння 9. Увести рядок алфавітних та цифрових символів. Видалити з рядка символи, що стоять на парних позиціях. Подвоїти усі алфавітні символи, що введені у верхньому регістрі. Цифрові символи замінити на номер студентської групи. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати методи класу **Math**   1. Згенерувати натуральні значення елементів одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом сортування вставками. Вивести на консоль масив до та після сортування. 2. Перевірити, чи елементи згенерованого масиву з парними індексами є простими числами. Для визначення простих чисел використати алгоритм Ератосфена, Якщо такі числа знайдені, вивести їх на консоль, інакше вивести відповідне повідомлення. 3. Переставити елементи масиву так, щоб спочатку розташовувалися прості числа, потім складені, кратні 3, потім складені кратні 2 і 3, потім усі інші. Вивести на консоль масив після переставлення елементів. 4. Знайти елементи масиву, що є квадратами цілих чисел, застосувавши алгоритм лінійного пошук. Вивести знайдені значення та їх індекси на консоль. 5. Визначити кількість і значення елементів масиву, що належать заданому з консолі діапазону, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів з консолі. На перетині i-го рядка та j-го стовпчика матриці записана кількість студентів, що проходять практику в j-му місяці в i-й компанії. Вивести матрицю на консоль. Визначити загальну кількість студентів, що проходили практику у всіх компаніях в заданому з консолі місяці, кількість компаній, в яких проходили практику задана з консолі кількість студентів, компанію, в якій практикувалась найбільша кількість студентів. Результати вивести на консоль. 7. Відсортувати кожний рядок згенерованої матриці за зростанням значень її елементів, визначити суму елементів рядків та переставити їх відповідно до спадання сум елементів по рядках. Вивести перетворену матрицю на консоль. 8. Знайти корені нелінійного рівняння (x^2−5\*x+7)^2−(x−2)(x−3)=0, застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з клавіатури рядок алфавітних, цифрових і символів знаків арифметичних дій. З рядка видалити всі алфавітні символи. Перевести цифрові символи у числові типи та вивести числа на екран. Над отриманими числами виконати арифметичні дії відповідно до заданих в рядку операцій. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати методи класу **Math**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом сортування вибором. Вивести на консоль масив до та після сортування. 2. Перевірити, чи непарні елементи згенерованого масиву є простими числами. Для визначення простих чисел використати алгоритм Ератосфена, Якщо такі числа знайдені, вивести їх на консоль, інакше вивести відповідне повідомлення. 3. Переставити елементи масиву так, щоб спочатку розташовувалися прості числа, потім складені, які діляться на 2, але не діляться на 3, потім числа, що діляться на 2 та на 3, потім числа, що діляться тільки на 3, потім усі інші. Вивести на консоль масив після перестановки елементів. 4. Знайти в масиві та вивести на консоль елементи та їх індекси, які є п’ятикутними числами, застосувавши алгоритм лінійного пошуку. Формула для обчислення *n*-го п’ятикутного числа  (1, 5, 12, 22,…). 5. Визначити усі входження в масив заданого з консолі елемента, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Вивести на консоль індекси шуканого елемента. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність з консолі. Матриця означує дисципліни та компетентності, які вони формують, і містить значення 0 або 1. На перетині i‑го рядка та j-го стовпчика матриці записаний 0, якщо дисципліна не забезпечує формування компетентності студента, 1 – якщо між дисципліною і компетентністю є зв’язок. Визначити, кількість дисциплін, які не забезпечують формування жодної компетентності студентів, дисципліни, які формують найбільшу кількість компетентностей, компетентності, які не забезпечені дисциплінами. Результати вивести на консоль. 7. В згенерованій матриці визначити суми елементів в стовпчиках, записати їх в одновимірний масив, відсортувати його, переставити стовчики матриці відповідно до зростання їх сум. 8. Знайти корені нелінійного рівняння x^3+6x+4x^2+3=0, застосувавши метод Ньютона. Результати вивести на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння 9. Увести з клавіатури рядок символів, включаючи розділові символи (пробіли, коми та крапки) та круглі дужки. Знайти слова, які мають парну довжину, та підрахувати їх кількість. Видалити символи, що розміщені в круглих дужках. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати методи класу **Math**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом сортування Шелла. Вивести на консоль масив до та після сортування. 2. Знайти в згенерованому масиві прості числа, застосувавши рещето Сундарама. Якщо такі числа знайдені, вивести їх на консоль, інакше вивести відповідне повідомлення. Посилання на алгоритм решета Сундарама: <https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%88%D0%B5%D1%82%D0%BE_%D0%A1%D1%83%D0%BD%D0%B4%D0%B0%D1%80%D0%B0%D0%BC%D0%B0> 3. Переставити елементи масиву так, щоб на парних позиціях розмістилися елементи з лівої частини відсортованого масиву з індексами від 0 до серединного елемента-1, на непарних позиціях розмістилися елементи з правої частини відсортованого масиву з індексами від кінцевого елемента до серединного. Вивести на консоль масив після перестановки елементів. 4. Знайти в масиві та вивести на консоль елементи та їх індекси, які є пірамідальними числами, застосувавши алгоритм лінійного пошуку. Формула для обчислення n-го пірамідального числа (1, 4, 10, 35,…): . 5. Визначити усі входження в масив заданого з консолі елемента, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Вивести на консоль індекси шуканого елемента. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів з консолі. На перетині i-го рядка та j-го стовпчика матриці записана кількість кредитів, що має i-та дисципліна в j-му навчальному році. Якщо значення елемента матриці дорівнює нулю, це означає, що i-та дисципліна не вивчається в j-му навчальному році Вивести матрицю на консоль. Визначити загальну кількість кредитів по всіх дисциплінах за усі навчальні роки, дисципліну та навчальний рік, в якому є дисципліна з найбільшою кількістю кредитів, середню кількість кредитів в кожному навчальному році. Результати вивести на консоль. 7. В згенерованій матриці знайти рядок або стовпчик з найбільшою кількістю нульових елементів, видалити відповідний рядок або стовпчик. Вивести на консоль перетворену матрицю. 8. Знайти корені нелінійного рівняння (x−1)^3+(2x+3)^3=27x^3+8, застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів, включаючи розділові (пробіли, коми та крапки). Визначити в рядку та вивести на консоль слова і підрахувати їх кількість. Слова – це лексеми, як не містять всередині розділових символів. Кожне друге слово розбити на склади. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом швидкого сортування. Вивести на консоль масив до та після сортування. 2. Згенерувати другий масив простих чисел від 1 до n, задавши значення n з консолі. Для отримання простих чисел використати решето Сундарама. Об’єднати масив простих чисел із згенерованим раніше масивом, не порушуючи його впорядкованість за зростанням. Надрукувати прості числа та об’єднаний масив. Посилання на (<https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%88%D0%B5%D1%82%D0%BE_%D0%A1%D1%83%D0%BD%D0%B4%D0%B0%D1%80%D0%B0%D0%BC%D0%B0>). 3. Переставити елементи відсортованого масиву так, щоб елементи з парними та непарними індексами помінялися місцями. Вивести на консоль масив після перестановки елементів. 4. Знайти в масиві та вивести на консоль елементи та їх індекси, які є шестикутними числами, застосувавши алгоритм лінійного пошуку. Шестикутні числа складають послідовність (1, 6, 15, 28,…). Формула для обчислення *n*-го шестикутного числа: 5. Знайти в масиві та вивести на консоль елементи та їх індекси, які є шестикутними числами, застосувавши алгоритм бінарного пошуку. Шестикутні числа складають послідовність (1, 6, 15, 28,…). Формула для обчислення *n*-го шестикутного числа: . Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch() класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів з консолі. На перетині *i*-го рядка та *j*-го стовпчика матриці записана кількість студентів, які працюють дорівнює *i*-ій компанії в *j*-му навчальному році. Якщо значення елемента матриці нулю, це означає, що в *i-*ій компанії не працює жодного студента в *j*-му навчальному році Вивести матрицю на консоль. Визначити компанію і рік, коли працюючих студентів найбільше, середню кількість працюючих студентів за усі роки в кожній компанії та компанії, в яких не працює жодного студента. 7. Із згенерованої матриці видалити компанії, в яких не працює жодного студента, відсортувати рядки матриці за зростанням, здійснити пошук в матриці введеного з консолі значення. Результати вивести на консоль. 8. Знайти корені нелінійного рівняння x4−13x2+36=0, застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести рядок символів, включаючи розділові (пробіли, коми, крапки). Слова є послідовністю символів без пробілів. Розбити рядок на слова, визначити їх кількість, знайти слово, що має найбільшу кількість голосних. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом бульбашкового сортування. Вивести на консоль масив до та після сортування. 2. Знайти в згенерованому масиві прості числа, застосувавши рещето Ератосфена. Якщо такі числа знайдені, вивести їх на консоль, інакше вивести відповідне повідомлення. 3. В масиві знайти мінімальний та максимальні елементи. Поміняти їх місцями. Вивести на консоль масив після перестановки. 4. Знайти в масиві та вивести на консоль елементи та їх індекси, які є семикутними числами, застосувавши алгоритм лінійного пошуку. Семикутні числа складають послідовність (1, 7, 18, 34,…). Формула для обчислення *n*-го семикутного числа: 5. Визначити усі входження в масив заданого з консолі елемента, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Вивести на консоль індекси шуканого елемента. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів з консолі. На перетині *i*-го рядка та *j*-го стовпчика матриці записана кількість студентів, які вибрані до складу студради *i*-го факультету в *j*-му навчальному році. Вивести матрицю на консоль. Визначити факультет і навчальний рік, в якому кількість студентів в студраді є найбільшою, сумарну кількість студентів в студраді за усі роки навчання на заданому з консолі факультеті, навчальний рік, в якому сумарна кількість студентів в студрадах усіх факультетів найменша. Результати вивести на консоль. 7. Із згенерованої матриці видалити факультети, на яких в заданому з консолі навчальному році студрада мала найменшу кількість, , відсортувати рядки матриці за зростанням, здійснити пошук в матриці введеного з консолі значення. Результати вивести на консоль. 8. Знайти корені нелінійного рівняння , застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з клавіатури рядок символів, який містить слова та розділові символи. Видалити з рядка однолітерні слова та зайві пропуски, коми, крапки. Вивести перетворений рядок на консоль. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом сортування включенням. Вивести на консоль масив до та після сортування. 2. Знайти в згенерованому масиві прості числа, застосувавши рещето Сундарама. Якщо такі числа знайдені, вивести їх на консоль, інакше вивести відповідне повідомлення. Посилання на алгоритм решета Сундарама: <https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%88%D0%B5%D1%82%D0%BE_%D0%A1%D1%83%D0%BD%D0%B4%D0%B0%D1%80%D0%B0%D0%BC%D0%B0> 3. В масиві знайти мінімальний та максимальні елементи. Елементи, що розташовані між мінімальним та максимальним записати в інвертованому порядку. Вивести результати на консоль. 4. Знайти в масиві та вивести на консоль елементи та їх індекси, які є семикутними числами, застосувавши алгоритм лінійного пошуку. восьмикутні числа складають послідовність (1, 8, 21, 40,…). Формула для обчислення *n*-го восьмикутного числа: 5. Визначити усі входження в масив восьмикутних чисел, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Вивести на консоль індекси шуканого елемента. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів з консолі. На перетині *i*-го рядка та *j*-го стовпчика матриці записана кількість заяв абітурієнтів, які подали заяви до вступу на *i*-й факультет та *j*-ту спеціальність. Вивести матрицю на консоль. Визначити факультет і спеціальність, на яку подана найбільші кількість заяв абітурієнтів, сумарну кількість заяв на усі спеціальності заданого з консолі факультету, сумарну кількість заяв на задану з консолі спеціальність на всіх факультетах. Результати вивести на консоль. 7. Із згенерованої матриці видалити факультети, на яких сумарна кількість заяв на всі спеціальності найменша. Переставити стовпці матриці в порядку спадання сумарної кількості заяв на спеціальність. Здійснити пошук в матриці введеного з консолі значення. Результати вивести на консоль. 8. Знайти корені нелінійного рівняння , застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести рядок символів, який складається із круглих дужок і знаків запитання. Передбачити вкладеність дужок, наприклад, "(???(???(?)??)". замінити знаки запитання круглими дужками так, щоб отримати правильний дужковий вираз, в якому кількість правих дужок дорівнює кількості лівих дужок, та порахувати кількість способів таких замін. Вивести перетворений рядок на консоль. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом сортування вибором. Вивести на консоль масив до та після сортування. 2. Знайти прості числа в заданому з консолі діапазоні, використавши решето Ератосфена. Записати прості числа в одновимірний масив. Об’єднати масив простих чисел із згенерованим раніше, виключивши повторення елементів і зберігаючи їх упорядкованість. Вивести масив на консоль. 3. У створеному раніше масиві поміняти місцями ліву та праву частини. В якості розділового елемента вибрати серединний елемент. Вивести результати на консоль 4. Знайти в масиві та вивести на консоль елементи та їх індекси, які є семикутними числами, застосувавши алгоритм лінійного пошуку. десятикутні числа складають послідовність (1, 10, 27, 52,…). Формула для обчислення *n*-го семикутного числа: 5. Визначити усі входження в масив семикутних чисел, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Вивести на консоль індекси шуканого елемента. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів з консолі. На перетині *i*-го рядка та *j*-го стовпчика матриці записана кількість опадів, що випали в *i*-му місяці *j*-то року. Вивести матрицю на консоль. Визначити рік та місяць, в якому випала найбільша кількість опадів, сумарну кількість опадів, що випали у зимовий період за усі роки, кількість опадів, що випала в заданий з консолі рік. Результати вивести на консоль. 7. Із згенерованої матриці видалити роки із найменшою кількістю опадів за усі місяці. Переставити рядки матриці в порядку спадання сумарної кількості опадів за усі роки. Здійснити пошук в матриці введеного з консолі значення. Результати вивести на консоль. 8. Знайти корені нелінійного рівняння , застосувавши метод Ньютона Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів, який складається зі слів та пропусків між ними. Визначити кількість повторів кожного слова та саме слово, що зустрічається найчастіше в тексті. Результати вивести на консоль. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом сортування Шелла. Вивести на консоль масив до та після сортування. 2. Знайти прості числа, задавши верхню границю чисел і використавши решета Сундарама. Посилання на опис алгоритму <https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%88%D0%B5%D1%82%D0%BE_%D0%A1%D1%83%D0%BD%D0%B4%D0%B0%D1%80%D0%B0%D0%BC%D0%B0>  Записати знайдені прості числа у одновимірний масив та вивести його на консоль. 3. У створеному раніше масиві перший і останній, максимальний і мінімальний елементи місцями. Результати вивести на консоль. 4. Знайти в масиві та вивести на консоль елементи та їх індекси, які є центрованими k‑кутними числами, застосувавши алгоритм лінійного пошуку. Формула для обчислення *n*-го центрованого k‑кутного числа: . Значення k увести з консолі. 5. Визначити усі входження в масив центрованих k‑кутних чисел, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Вивести на консоль індекси шуканого елемента. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array. 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів з консолі. На перетині *i*-го рядка та *j*-го стовпчика матриці записана оцінка *i*-го студента по *j*-й дисципліні. Вивести матрицю на консоль. Визначити студентів, які за підсумками успішності за всіма дисциплінами підпадають під відрахування, рейтинг кожного студента як середнє арифметичне по всіх дисциплінах, дисципліну, середній бал якої найбільший. Результати вивести на консоль. 7. Із згенерованої матриці видалити дисципліну, середній бал якої найменший по всіх студентах. Переставити стовпці матриці в порядку спадання середнього балу за дисциплінами. Здійснити пошук в матриці введеного з консолі значення. Результати вивести на консоль. 8. Знайти корені нелінійного рівняння x^3−4x+6=0, застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів, який складається зі слів та пропусків між ними. Створити новий рядок, в якому слова відсортовані за спаданням довжини слів. Відсортувати рядок за алфавітом першої літери слів. Результати вивести на консоль. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом швидкого сортування. Вивести на консоль масив до та після сортування. 2. У згенерованому масиві визначити прості числа, застосувавши решето Ератосфена, і надрукувати їх. Якщо прості числа не знайдені, вивести відповідне повідомлення. 3. У створеному масиві переставити елементи так, щоб в середині масиву знаходився мінімальний елемент, ліва частина, рахуючи від середини масиву, упорядкована за спаданням, права частина упорядкована за зростанням. Результати перетворення масиву вивести на консоль. 4. Знайти в масиві та вивести на консоль елементи та їх індекси, які є центрованими квадратними числами, застосувавши алгоритм лінійного пошуку. Формула для обчислення *n*-го центрованого квадратного числа: . 5. Визначити усі входження в масив центрованих квадратних чисел, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Вивести на консоль індекси шуканого елемента. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array. 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів з консолі. На перетині *i*-го рядка та *j*-го стовпчика матриці записана кількість підручників, які *i*-й студент прочитав по *j*-й дисципліні. Вивести матрицю на консоль. Визначити студентів, які прочитали найбільшу кількість підручників за всіма дисциплінами, дисципліну, по якій рекомендована найменша кількість підручників, сумарну кількість підручників, які прочитали усі студенти по всіх дисциплінах. Результати вивести на консоль. 7. Із згенерованої матриці видалити студента, який прочитав найменшу кількість підручників за всіма дисциплінами. Переставити стовпці матриці в порядку спадання сумарної кількості підручників, які прочитали всі по кожній дисципліні. Результати вивести на консоль. 8. Знайти корені нелінійного рівняння , застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів. Підрахувати кількість повторів кожного символу, видалити символи, що повторюються. Результати вивести на консоль. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math.**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом бульбашкового сортування. Вивести на консоль масив до та після сортування. 2. У згенерованому масиві визначити прості числа, застосувавши решето Сундарама. Посилання на опис алгоритму <https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D1%88%D0%B5%D1%82%D0%BE_%D0%A1%D1%83%D0%BD%D0%B4%D0%B0%D1%80%D0%B0%D0%BC%D0%B0>.  Вивести прості числа на консоль. Якщо прості числа не знайдені, вивести відповідне повідомлення. 3. У створеному масиві визначити максимальний елемент серед елементів з парними індексами, і мінімальний елемент серед елементів з непарними індексами. Вивести знайдені елементи та їх індекси на консоль. Поміняти місцями знайдені елементи. Використати методи класу Math. Результати перетворення масиву вивести на консоль. 4. Знайти у створеному масиві та вивести на консоль елементи та їх індекси, які є числами Фібоначчі, що визначаються за формулою F1 = 1; F2 = 1; Fk = Fk-1 + Fk-2 для k>2. 5. Визначити усі входження у створений масив центрованих трикутних чисел, застосувавши метод бінарного пошуку. Формула для обчислення n-го центрованого трикутного числа: . У разі їх відсутності вивести відповідне повідомлення. Вивести на консоль індекси шуканих елементів. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array. 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів (0…15) з консолі. На перетині *i*-го рядка та *j*-го стовпчика матриці записана кількість лабораторних робіт, виконаних *i*-м студент з *j*-ї дисципліни. Вивести матрицю на консоль. Визначити студентів, які виконали найбільшу кількість лабораторних робіт за всіма дисциплінами, дисципліну, по якій усі студенти виконали найменшу кількість робіт, студентів, які хоча б з одної дисциплінами не виконали жодної (в матриці записаний 0) лабораторної роботи. Результати вивести на консоль. 7. Із згенерованої матриці видалити студентів (рядки), які хоча б з одної дисциплінами не виконали жодної (в матриці записаний 0) лабораторної роботи. Вивести перетворену матрицю на консоль. Якщо таких студентів немає, вивести відповідне повідомлення. 8. Знайти корені нелінійного рівняння , застосувавши метод Ньютона. Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів. Видалити слова, які складаються з одної літери. Підрахувати кількість слів, перша і остання літера яких однакові. Якщо таких слів нема, вивести відповідне повідомлення. Результати вивести на консоль. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math.**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом швидкого сортування. Вивести на консоль масив до та після сортування. 2. В згенерованому масиві визначити прості числа, застосувавши решето Ератосфена, і надрукувати їх. Якщо прості числа не знайдені, вивести відповідне повідомлення. 3. У створеному масиві знайти досконалі числа, вивести їх значення та індекси на консоль. Досконале число – це число, яке дорівнює сумі всіх своїх дільників, у тому числі одиниця, але виключаючи саме себе, наприклад число 6 – досконале, оскільки його дільники 1,2,3 в сумі складають 6. Застосувати метод лінійного пошуку. 4. Знайти в масиві та вивести на консоль елементи та їх індекси, які є числами Мерсенна, застосувавши алгоритм лінійного пошуку. Формула для обчислення *n*-го числа Мерсенна: . 5. Визначити усі входження в масив заданого з консолі елемента, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Вивести на консоль індекси шуканого елемента. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів з консолі. На перетині *i*-го рядка та *j*-го стовпчика матриці записана кількість пропущених занять *i*-м студент з *j*-ї дисципліни. Вивести матрицю на консоль. Визначити студентів, які пропустили найбільшу кількість занять за всіма дисциплінами, дисципліну, по якій пропущена всіма студентами найменша кількість занять, середнє значення пропусків всіма студентами занять з усіх дисциплін. Результати вивести на консоль. 7. Із згенерованої раніше матриці видалити студентів (рядки), які пропустили найбільшу кількість занять. Вивести перетворену матрицю на консоль. 8. Знайти корені нелінійного рівняння, застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів, який містить алфавітні та розділові символи (пробіли, точки, коми). Визначити кількість слів в рядку. Знайти найдовше слово. Першу літеру слова, що записано після точки, перевести у верхній регістр. Результати вивести на консоль. |
| 1. **Й** | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math.**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом сортування вибором. Вивести на консоль масив до та після сортування. 2. Визначити прості числа, задавши найбільшу межу чисел і застосувавши решето Ератосфена. Записати прості числа у масив і вивести його на консоль. Дописати в кінець відсортованого раніше масиву масив простих чисел. 3. Знайти в масиві та вивести на консоль елементи та їх індекси, які є числами Фібоначчі, Мерсенна () або Ферма (), застосувавши алгоритм лінійного пошуку. Якщо вказаних чисел в масиві не знайдено, вивести відповідне повідомлення. 4. У створеному масиві переставити елементи так, щоб спочатку йшли числа Фібоначчі, потім числа Мерсенна, потім числа Ферма, потім решта. Результати перетворення масиву вивести на консоль. Застосувати метод лінійного пошуку. Якщо вказаних чисел в масиві не знайдено, визначити середнє арифметичне та середнє геометричне елементів масиву. 5. Визначити усі входження в масив заданого з консолі елемента, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Вивести на консоль індекси шуканого елемента. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів з консолі, включаючи нульове значення. На перетині *i*-го рядка та *j*-го стовпчика матриці записана кількість резюме, посланих *i*-им студентом для працевлаштування в *j*-ту компанію. Вивести матрицю на консоль. Визначити студентів, які послали резюме у всі компанії, компанію, в яку послали найменшу кількість резюме, студента, який не послав резюме в жодну компанію. Результати вивести на консоль. 7. Із згенерованої раніше матриці видалити студента, який не послав жодного резюме в компанію. Вивести перетворену матрицю на консоль. Якщо такого студента немає, вивести відповідне повідомлення. 8. Знайти корені нелінійного рівняння , застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів, який містить алфавітні та розділові символи (пробіли, точки, коми). Визначити кількість слів в рядку. Упорядкувати слова так, щоб вийшов "ланцюжок" слів, коли перша буква поточного слова збігається з останньою буквою попереднього слова, а остання буква поточного слова – з першою буквою наступного слова. Якщо ланцюжок створити неможливо, вивести відповідне повідомлення. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math.**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом сортування Шелла. Вивести на консоль масив до та після сортування. 2. У згенерованому масиві визначити прості числа, застосувавши решето Ератосфена, і надрукувати їх. Якщо прості числа не знайдені, вивести відповідне повідомлення. 3. У створеному масиві переставити елементи так, щоб спочатку йшли парні елементи, якщо їх індекси у вхідному масиві парні, потім непарні елементи, якщо їх індекси у вхідному масиві непарні, потім елементи з нульовим значенням, потім решта. Результати перетворення масиву вивести на консоль. 4. Знайти в масиві та вивести на консоль елементи та їх індекси, які є квадратами чисел, застосувавши алгоритм лінійного пошуку. 5. Визначити усі входження в масив заданого з консолі елемента, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Вивести на консоль індекси шуканого елемента. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів з консолі. На перетині *i*-го рядка та *j*-го стовпчика матриці записана температура повітря в *i*-ий день *j*-го місяця. Вивести матрицю на консоль. Визначити місяці, середня температура в яких була вище за задану з консолі, день, в якому температура була найнижча по всіх місяцях, середню температура взимку. Результати вивести на консоль. 7. У згенеровану раніше матрицю додати рядок, який для кожного місяця буде містити середнє значення температури повітря за місяць. Вивести перетворену матрицю на консоль. 8. Знайти корені нелінійного рівняння , застосувавши метод половинного ділення (метод бісекції). Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів, який містить алфавітні та розділові символи (пробіли, точки, коми). Визначити кількість слів в рядку. Якщо слово починається з голосної, його першу літеру перевести у верхній регістр. Великі літери в середині слова перевести у нижній регістр. Результати вивести на консоль. |
|  | Створити консольний застосунок мовою C#. Використати методи класів **Console**, **Convert** в процесі введення та виведення даних. Реалізувати перераховані функції. Виклик функцій здійснити за допомогою меню, застосувавши оператор вибору **switch**. Для створення масиву і матриці використати методи класу **Random**. Для виконання математичних обчислень використати клас **Math.**   1. Згенерувати елементи одновимірного масиву, задавши їх кількість та діапазон значень з консолі. Відсортувати згенерований масив за зростанням значень його елементів алгоритмом сортування включенням. Вивести на консоль масив до та після сортування. 2. У згенерованому масиві визначити прості числа, застосувавши решето Ератосфена, і надрукувати їх. Якщо прості числа не знайдені, вивести відповідне повідомлення. 3. У створеному масиві для кожної пари сусідніх елементів знайти найбільший спільний дільник і записати їх в новий одновимірний масив. Отриманий масив вивести на консоль. 4. Знайти в масиві та вивести на консоль елементи та їх індекси, які є центрованими трикутними числами, застосувавши алгоритм лінійного пошуку. Формула для обчислення *n*-го центрованого трикутного числа: . Якщо центровані трикутні числа не знайдені, вивести відповідне повідомлення. 5. Визначити усі входження в масив заданого з консолі елемента, застосувавши метод бінарного пошуку. У разі їх відсутності вивести відповідне повідомлення. Вивести на консоль індекси шуканого елемента. Модифікувати функцію бінарного пошуку у масиві, використавши метод BinarySearch класу Array 6. Згенерувати елементи матриці, задавши її вимірність та діапазон значень елементів з консолі, включаючи нульове значення. *i*-й рядок означує телеканал, *j*-й стовпчик означує телепрограму. На перетині *i*-го рядка та *j*-го стовпчика матриці записана кількість телепрограм. Вивести матрицю на консоль. Визначити телеканал, який має найбільшу кількість телепрограм, телепрограму, яку транслюють усі телеканали, середню кількість телепрограм по всіх телеканалам. Результати вивести на консоль. 7. Із згенерованої раніше матриці видалити рядки, номери яких задані з консолі і свідчать про закриття телеканалів. Вивести на консоль перетворену матрицю. 8. Знайти корені нелінійного рівняння , застосувавши метод Ньютона. Вивести результати на консоль. Здійснити перевірку правильності рішення, підставивши знайдені значення коренів в нелінійне рівняння. 9. Увести з консолі рядок символів, який містить алфавітні, цифрові та розділові символи (пробіли, точки, коми). Видалити з рядка усі символи, які не є буквами. Визначити кількість слів, які починаються та закінчуються на задану з консолі букви. слів в рядку. Результати вивести на консоль. |

### Література

1. О.С.Бичков, Є.В.Іванов Об’єктно-орієнтоване програмування мовою C#/

2.C# 2005 и платформа .Net 3.0 для профессионалов. Нейгел К., Ивьен Б. и др. –М.: ООО “И.Д. Вильямс", 2008. –1376 с.