

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

**Л.В. Соловей, Н.М. Мірошніченко, Т.Г. Бабак**

## **ПРОГРАМУВАННЯ МОВОЮ C# Visual Studio 2019**

### **Лабораторний практикум**

для студентів хімічних спеціальностей усіх форм навчання

У трьох частинах

### **Частина 2**

ЗАТВЕРДЖЕНО  
редакційно-видавничою  
радою університету,  
протокол № 1 від 28.01.2022 р.

Харків  
НТУ «ХПІ»  
2022

УДК 044.43 (076.5)

С 60

Р е ц е н з е н т и:

*В. М. Колодяжний*, д-р фіз.-мат. наук, професор ХНАДУ;

*Є. Д. Пономаренко*, доцент НТУ «ХП»

Соловей Л.В.

С 60 Програмування мовою С# Visual Studio 2019. Лабораторний практикум: у 3-х ч. Ч. 2. / Л.В. Соловей, Н. М. Мірошніченко, Т. Г. Бабак. – Харків : НТУ «ХП», 2022. – 108 с.

Лабораторний практикум присвячений вивченню мови програмування С# Visual Studio.NET 2019. Наведено велику кількість прикладів написання програм різної складності. До прикладів надаються пояснення. Усі програми забезпечені результатами виконання. До кожної теми подані практичні завдання для виконання лабораторних робіт. Матеріал розміщено за принципом поступовості, починаючи з основ до більш складних можливостей мови програмування С#.

Призначено для студентів хімічних спеціальностей, у тому числі для іноземних студентів.

Ил. 52    Табл. 2    Бібліогр. 3

**УДК 044.43 (076.5)**

© Соловей Л.В., Мірошніченко Н.М., Бабак Т.Г., 2022

© НТУ «ХП», 2022

## ЗМІСТ

### Рішення задач у вигляді консольних додатків

<b>Тема 6. МЕТОДИ</b> .....	4
6.1 Приклади розв'язання задач .....	4
6.1.1. Методи-функції .....	4
6.1.2. Методи-процедури .....	16
6.2 Завдання для виконання лабораторних робіт .....	42
<b>Тема 7. РЯДКИ</b> .....	45
7.1 Приклади розв'язання задач .....	45
7.2 Завдання для виконання лабораторних робіт .....	54
<b>Тема 8. КЛАСИ</b> .....	58
8.1 Приклади розв'язання задач .....	59
8.2 Завдання для виконання лабораторних робіт .....	83
<b>Тема 9. СТРУКТУРИ, ПЕРЕРАХУВАННЯ</b> .....	84
9.1 Приклади розв'язання задач .....	84
9.2 Завдання для виконання лабораторних робіт .....	91
<b>Тема 10 ФАЙЛИ</b> .....	92
10.1 Приклади розв'язання задач .....	92

## Тема 6. МЕТОДИ

*Мета роботи:* вивчити синтаксис методу, типи параметрів методу, способи виклику методів.

### 6.1. Приклади розв'язання задач

#### 6.1.1. Методи-функції

У багатьох методах усі величини, що метод повинен отримати як вхідні дані, описуються в списку параметрів, а величина, яку обчислює метод як результат своєї роботи, повертається до викликаючого коду за допомогою оператора `return`.

**Задача 6.1.** Написати метод для обчислення середнього арифметичного трьох дійсних чисел.

*Програмний код:*

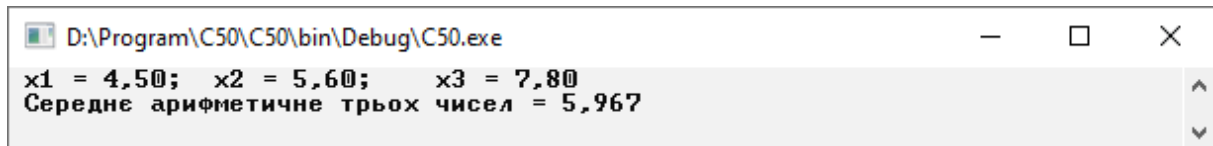
```
using System;

namespace C50
{
    class Program
    {
        static double Aver(double x, double y, double z)
        {
            return (x + y + z) / 3;
        }

        static void Main(string[] args)
        {
            double res, x1, x2, x3;
            x1 = 4.5; x2 = 5.6; x3 = 7.8;
            res = Aver(x1, x2, x3);
            Console.WriteLine($" x1 = {x1:F2}; x2 = {x2:F2};
                               x3 = {x3:F2}");
            Console.WriteLine($" Середнє арифметичне трьох чисел =
                               {res:F3}");

            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 6.1.



```
D:\Program\C50\C50\bin\Debug\C50.exe
x1 = 4.50; x2 = 5.60; x3 = 7.80
Середнє арифметичне трьох чисел = 5.967
```

Рисунок 6.1 – Результати розв’язання задачі 6.1

*Пояснення:*

`static double Aver(double x, double y, double z)` – це заголовок методу;

`Aver` – ім’я методу;

`(double x, double y, double z)` – параметри-значення.

Фактично обчислення середнього арифметичного значення з трьох заданих чисел здійснюється з тіла головного методу `Main()` за допомогою виклику методу `Aver`:

```
res = Aver(x1, x2, x3);
```

При виклику цього методу передаються як аргументи (*фактичні параметри*) змінні `x1`, `x2`, `x3` типу `double`. Відповідно, у визначенні методу `Aver` ідентифікатори `x`, `y` і `z` називаються формальними параметрами.

**Задача 6.2.** Написати метод, який обчислює площу поверхні паралелепіпеда  $S = (L \cdot w + L \cdot h + w \cdot h) \cdot 2$

*Програмний код:*

```
using System;

namespace C52a
{
    class Program
    {
        // площа поверхні паралелепіпеда
        static double ps(double L, double w, double h)
        {
            return (L * w + L * h + w * h) * 2;
        }

        static void Main(string[] args)
        {
            double L, w, h; // довжина, ширина і висота паралелепіпеда
            double s; // площа поверхні паралелепіпеда
            Console.WriteLine(" Обчислення площі поверхні
                                паралелепіпеда \n ");
            Console.WriteLine(" Введіть початкові дані:");
            Console.Write(" Довжина (см) ? ");
        }
    }
}
```

```

        L = Convert.ToDouble(Console.ReadLine());
        Console.Write(" Ширина (см) ? ");
        w = Convert.ToDouble(Console.ReadLine());
        Console.Write(" Висота (см) ? ");
        h = Convert.ToDouble(Console.ReadLine());
        s = ps(L, w, h);
        Console.WriteLine($" \n Площа поверхні =
                                {s:F2} кв.см.");
        Console.ReadLine();
    }
}
}

```

Результати розрахунку подано на рис. 6.2.

```

D:\Program\C52a\C52a\bin\Debug\C52a.exe
Обчислення площі поверхні паралелепіпеда
Введіть початкові дані:
Довжина (см) ? 9
Ширина (см) ? 7,5
Висота (см) ? 5
Площа поверхні = 300,00 кв.см.

```

Рисунок 6.2 – Результати розв’язання задачі 6.2

*Пояснення:* Фактично обчислення площі поверхні паралелепіпеда здійснюється з тіла головного методу `Main()` за допомогою виклику методу `ps()`:

```
s = ps(L, w, h);
```

Як фактичні параметри при виклику цього методу передаються значення змінних `L`, `w`, `h` типу `double`. Відповідно, у визначенні методу `ps`:

```
static double ps(double L, double w, double h)
```

ідентифікатори `L`, `w`, `h` називаються *формальними параметрами*.

**Задача 6.3.** Написати метод перерахунку температури з градусів Фаренгейта в градуси Цельсія ( $^{\circ}\text{C} = 5/9 \cdot (^{\circ}\text{F} - 32)$ ) і програму, що використовує цей метод, яка виводить на екран таблицю відповідності температур у шкалах Фаренгейта і Цельсія.

*Програмний код:*

```

using System;

namespace C51
{
    class Program
    {

```

```

// перераховує температуру
// з градусів Фаренгейта в градуси Цельсія
static double ps(double f)
{
    double c;
    c = (5.0 / 9.0) * (f - 32);
    return (c);
}

static void Main(string[] args)
{
    double f;      // температура в градусах Фаренгейта
    double c;      // температура в градусах Цельсія
    double f1, f2; // діапазон зміни температури
    double df;     // крок зміни температури
    f1 = 32.0; f2 = 50.0; df = 2.0;
    Console.WriteLine("-----");
    Console.WriteLine("  F      C  ");
    Console.WriteLine("-----");
    for (f = f1; f <= f2; f = f + df)
    {
        c = ps(f);
        Console.WriteLine($"{f,5:F0} {c,5:F0}");
    }
    Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 6.3.

F	C
32	0
34	1
36	2
38	3
40	4
42	6
44	7
46	8
48	9
50	10

Рисунок 6.3 – Результати розв’язання задачі 6.3

**Задача 6.4.** Створити консольний додаток для обчислення виразу:

$$W = \sqrt{x^2 + y^2 + \sin^2 xy} + \sqrt{y^2 + z^2 + \sin^2 yz} + \sqrt{z^2 + x^2 + \sin^2 xz}$$

при  $x = 21,87$ ;  $y = 35,13$ ;  $z = 9,87$ .

У вигляді методу оформимо вираз  $\sqrt{a^2 + b^2 + \sin^2 ab}$ .

*Програмний код:*

```
using System;

namespace C59
{
    class Program
    {
        static double ps(double a, double b)
        {
            return Math.Sqrt(a*a + b*b + Math.Pow(Math.Sin(a*b), 2));
        }

        static void Main(string[] args)
        {
            double x, y, z, W;
            x = 21.87; y = 35.13; z = 9.87;
            W = ps(x, y) + ps(y, z) + ps(z, x);
            Console.WriteLine($" x= {x:F2}; y= {y:F2}; z = {z:F2}");
            Console.WriteLine($" W= {W:F3}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 6.4.



Рисунок 6.4 – Результати розв’язання задачі 6.4

**Задача 6.5.** Написати метод обчислення факторіала числа  $K!$ .

*Програмний код:*

```
using System;

namespace C53
{
```

```

class Program
{
    static double factorial(int K)
    {
        double p = 1;
        for (int i = 2; i <= K; i++)
            p = p * i;
        return (p);
    }

    static void Main(string[] args)
    {
        int N; double f;
        Console.Write(" Введіть N= ");
        N = Convert.ToInt32(Console.ReadLine());
        f = factorial(N);
        Console.WriteLine($" {N}!= {f} ");
        Console.ReadLine();
    }
}

```

Результати розрахунку подано на рис. 6.5.



Рисунок 6.5 – Результати розв’язання задачі 6.5

**Задача 6.6.** Написати метод для розрахунку суми цілих чисел від  $M$  до  $N$ .

*Програмний код:*

```

using System;

namespace C57
{
    class Program
    {
        static int MySum(int M, int N)
        {
            int i, sum;
            sum = 0;

```

```

        for (i = M; i <= N; i++)
            sum = sum + i;
        return sum;
    }

    static void Main(string[] args)
    {
        int M, N, S;
        Console.Write(" Введіть M= ");
        M = Convert.ToInt32(Console.ReadLine());
        Console.Write(" Введіть N= ");
        N = Convert.ToInt32(Console.ReadLine());
        S = MySum(M, N);
        Console.WriteLine($" Сума = {S}");
        Console.ReadLine();
    }
}

```

Результати розрахунку подано на рис. 6.6.

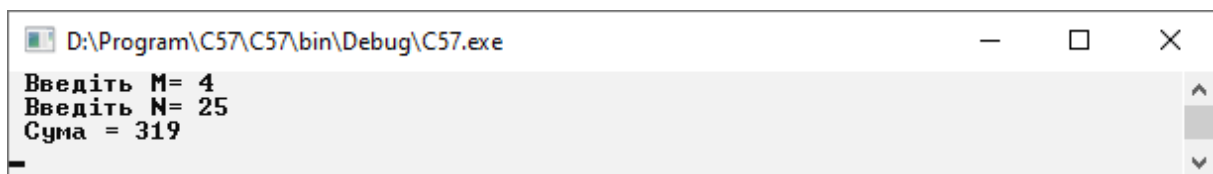


Рисунок 6.6 – Результати розв’язання задачі 6.6

**Задача 6.7.** Написати метод, який повертає середнє арифметичне елементів одновимірного масиву.

*Програмний код:*

```

using System;

namespace C58
{
    class Program
    {
        static double ps(int[] A)
        {
            int i;
            double sum = 0;
            for (i = 0; i < A.Length; i++)
                sum = sum + A[i];
            return sum / A.Length;
        }
    }
}

```

```

static void Main(string[] args)
{
    const int n = 15;    // кількість елементів масиву
    double sr;    // середнє арифметичне елементів масиву
    int[] A = new int[n] { 1, 3, 6, 7, 8, 9, 10, 12, 15, 17,
                          20, 22, 25, 28, 30 };

    Console.WriteLine(" Масив A ");
    for (int i = 0; i < n; i++)
        Console.WriteLine($" A[{i}]={A[i]}");
    sr = ps(A);
    Console.WriteLine($" Середнє арифметичне = {sr}");
    Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 6.7.

```

D:\Program\C58\C58\bin\Debug\C58.exe
Масив A
A[0]=1
A[1]=3
A[2]=6
A[3]=7
A[4]=8
A[5]=9
A[6]=10
A[7]=12
A[8]=15
A[9]=17
A[10]=20
A[11]=22
A[12]=25
A[13]=28
A[14]=30
Середнє арифметичне = 14.2

```

Рисунок 6.7 – Результати розв’язання задачі 6.7

*Пояснення:*

1) for(i=0; i < A.Length; i++)

Властивість `Length` (клас `Array`) – кількість елементів масиву.

Властивість `Length` дозволяє реалізовувати алгоритми, які будуть працювати з масивами різної довжини.

### Задача 6.8

1. Створити двовимірний масив  $A$  розміром  $4 \times 5$ ,

$$A = (a_{ij}; i = 0, 1, \dots, 3; j = 0, 1, \dots, 4).$$

Елементи масиву обчислити за формулою:

$$a_{ij} = 20 \cdot i - (0,5 \cdot j)^2 + 1.$$

2. Обчислити суму елементів масиву, причому обчислення суми елементів рядків масиву оформити у вигляді методу.

3. Відкоригувати програму для випадку, коли у вигляді методу потрібно оформити обчислення суми елементів стовпців (виконати самостійно).

*Програмний код:*

```
using System;

namespace C62
{
    class Program
    {
        static double ps(int i, double[,] B)
        {
            // Метод накопичує суму елементів за даним рядком
            int j;
            double c;
            c = 0;
            for (j = 0; j < B.GetLength(1); j++)
                c = c + B[i, j];
            return c;
        }

        static void Main(string[] args)
        {
            const int n = 4;    // кількість рядків
            const int m = 5;    // кількість стовпців
            double[,] A = new double[n, m];
            int i, j;
            double s;
            // Обчислення і виведення елементів масиву
            Console.WriteLine(" Масив A: ");
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < m; j++)
                {
                    A[i, j] = 20 * i - Math.Pow((0.5 * j), 2) + 1;
                    Console.Write($"{A[i,j],7:F2} ");
                }
            }
        }
    }
}
```

```

        Console.WriteLine();
    }
    Console.WriteLine();
    s = 0;
    for (i = 0; i < n; i++)
        s = s + ps(i, A);
    Console.WriteLine($" Сума елементів масиву= {s:F2}");
    Console.ReadLine();
}
}
}
}
}

```

Результати розрахунку подано на рис. 6.8.

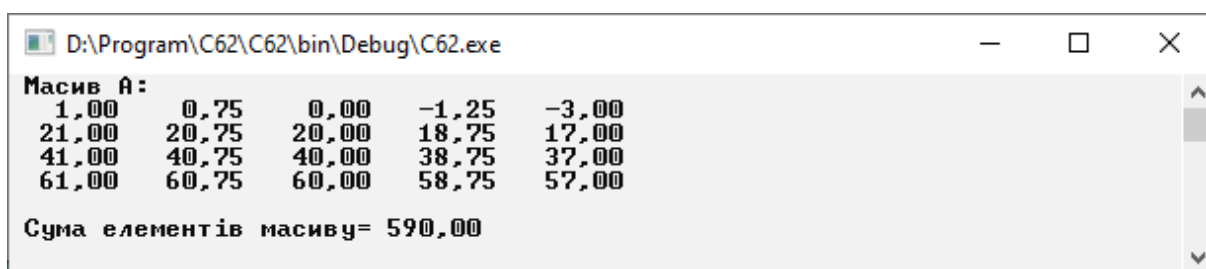


Рисунок 6.8 – Результати розв’язання задачі 6.8

**Задача 6.9.** Обчислити вираз:

$$F = \sum_{i=1}^5 (1,78 \cdot a_i + 5,4)^2 + \sum_{i=1}^8 (0,1 \cdot a_i + 1,25)^4 - \sum_{i=0}^{10} (8,2 \cdot a_i - 17,02)^3,$$

де  $a_i$  – елементи одновимірною масиву  $A(12)$ ;  $i = 0, 1, \dots, 11$ .

Обчислення суми виду  $\sum_{i=n1}^{n2} (k1 \cdot a_i + k2)^{k3}$  оформити у вигляді

методу.

Блок-схему розв’язання задачі подано на рис. 6.10.

*Програмний код:*

```

using System;

namespace C60
{
    class Program
    {
        static double ps(int n1, int n2, double k1, double k2,
double k3, double[] A)
        {
            int i;

```

```

        double s = 0;
        for (i = n1; i <= n2; i++)
            s = s + Math.Pow((k1 * A[i] + k2), k3);
        return s;
    }

    static void Main(string[] args)
    {
        const int n = 12; // Кількість елементів масиву
        double[] A = new double[n];
        int i;
        double r1, r2, r3, f;
        Console.WriteLine("Масив A: ");
        for (i = 0; i < n; i++)
        {
            A[i] = (i + 2) / 7.0;
            Console.WriteLine($" A[{i}]= {A[i]:F3}");
        }
        r1 = ps(1, 5, 1.78, 5.4, 2, A);
        r2 = ps(1, 8, 0.1, 1.25, 4, A);
        r3 = ps(0, 10, 8.2, -17.02, 3, A);
        f = r1 + r2 - r3;
        Console.WriteLine();
        Console.WriteLine($" f= {f:F3}");
        Console.ReadLine();
    }
}

```

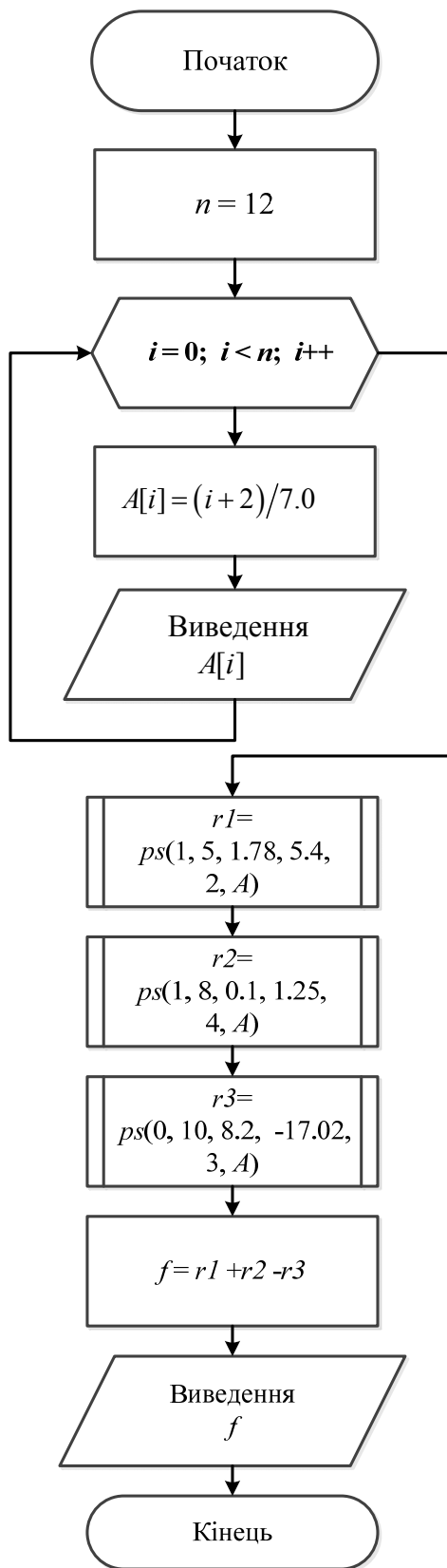
Результати розрахунку подано на рис. 6.9.

```

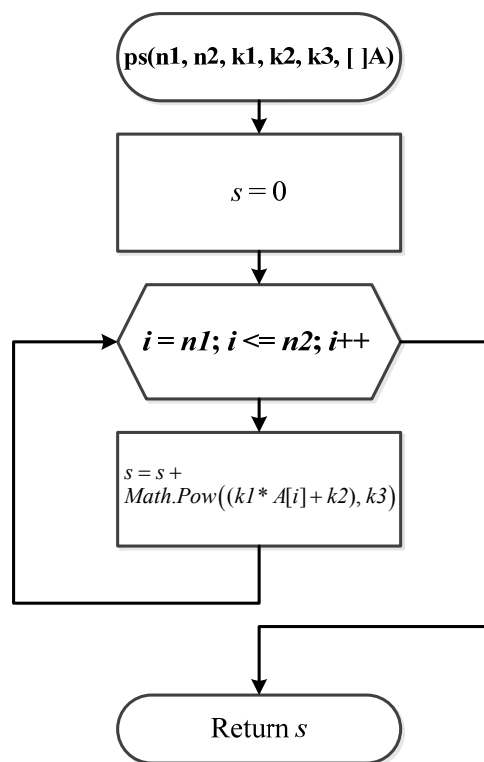
D:\Program\C60\C60\bin\Debug\C60.exe
Масив A:
A[0]= 0,286
A[1]= 0,429
A[2]= 0,571
A[3]= 0,714
A[4]= 0,857
A[5]= 1,000
A[6]= 1,143
A[7]= 1,286
A[8]= 1,429
A[9]= 1,571
A[10]= 1,714
A[11]= 1,857
f= 11790,768

```

Рисунок 6.9 – Результати розв’язання задачі 6.9



метод Main()



метод ps()

Рисунок 6.10 – Блок-схема алгоритму до задачі 6.9

## 6.1.2. Методи-процедури

Часто застосовуються методи, які не повертають значення (тип `void`), але дозволяють змінити в методі передані параметри. З цією метою використовуються параметри-посилання (з використанням ключового слова `ref`) і вихідні параметри (з використанням ключового слова `out`).

Параметри-посилання дозволяють змінити в методі переданий параметр, але параметр повинен мати значення при зверненні до методу.

Вихідні параметри дозволяють отримати в методі нове значення для параметра.

**Задача 6.10.** Створити метод, який обчислюватиме площу круга і довжину кола. Програмний викликаючий код повинен передавати методу як аргумент значення радіусу і друкувати значення радіусу, площі круга і довжини кола.

**Спосіб 1.** Використати параметри-посилання (описуються за допомогою ключового слова `ref`).

*Програмний код:*

```
using System;

namespace C63
{
    class Program
    {
        static void ps(double radius, ref double ploshad, ref double
dlina)
        {
            ploshad = Math.PI * radius * radius;
            dlina = 2 * Math.PI * radius;
        }

        static void Main(string[] args)
        {
            double r, s, L;
            Console.Write(" Введіть радіус r=? ");
            r = Convert.ToDouble(Console.ReadLine());
            // Перед викликом функції аргументам необхідно присвоїти значення
            s = 1; L = 1;
        }
    }
}
```

```

        ps(r, ref s, ref L);
        Console.WriteLine($" Площа круга = {s:F2} \n Довжина
                               кола = {L:F2}");

        Console.ReadLine();
    }
}
}

```

Результати розрахунку подано на рис. 6.11.



Рисунок 6.11 – Результати розв’язання задачі 6.10 (Спосіб 1)

*Пояснення:*

`static void ps(double radius, ref double ploshad, ref double dlina)` – це заголовок методу.

У списку параметрів методу є параметри-значення і параметри-посилання. Ознакою параметра-посилання є ключове слово `ref` перед описом параметра.

`double radius` – параметр-значення, він не може бути змінений функцією;

`ref double ploshad, ref double dlina` – параметри-посилання.

При виклику методу в ділянку параметрів копіюється не значення аргументу, а його адреса, і метод через нього має доступ до комірки, в якій зберігається аргумент. Таким чином, параметри-посилання передаються *за адресою* (частіше вживається термін «передача за посиланням»). Метод працює безпосередньо зі змінною з викликаючої функції і, отже, може її змінити, тому якщо в методі потрібно змінити значення параметрів, вони повинні передаватися тільки за посиланням.

Модифікатор `ref` вказується перед оголошенням параметра в самому методі і перед аргументом при виклику методу. Параметр типу `ref` має бути ініціалізований певним значенням до виклику методу. При виклику методу з головного методу `Main()` йому передається як вхідний параметр значення радіуса  $R = 5$ .

**Спосіб 2.** Використати вихідні параметри (описуються за допомогою ключового слова `out`).

*Програмний код:*

```

using System;

namespace C63a
{
    class Program
    {

```

```

    static void ps(double radius, out double ploshad, out double
dlina)
    {
        ploshad = Math.PI * radius * radius;
        dlina = 2 * Math.PI * radius;
    }

    static void Main(string[] args)
    {
        double r, s, L;
        Console.Write(" Введіть радіус r=? ");
        r = Convert.ToDouble(Console.ReadLine());
        ps(r, out s, out L);
        Console.WriteLine($" Площа круга = {s:F2} \n Довжина
                                                                    кола = {L:F2}");

        Console.ReadLine();
    }
}

```

Результати розрахунку подано на рис. 6.12.

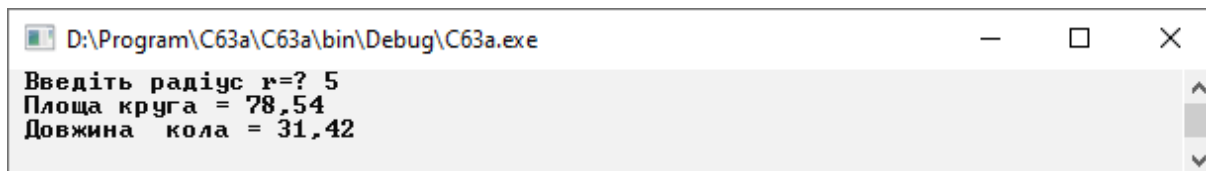


Рисунок 6.12 – Результати розв’язання задачі 6.10 (Спосіб 2)

*Пояснення:*

Методи можуть повертати у викликаючу програму одно або декілька значень за допомогою списку аргументів.

У даному випадку метод `ps()`

`static void ps(double radius, out double ploshad, out double dlina)` призначений для обчислення площі круга і довжини кола;

`ps` – ім’я методу, після імені в дужках вказуються параметри з потрібним типом даних. Серед них є вхідні параметри-значення і вихідні параметри. Ознакою вихідного параметра є ключове слово `out` перед описом параметра.

`double radius` – параметр-значення, він не може бути змінений функцією;

`out double ploshad, out double dlina` – вихідні параметри.

Модифікатор параметра `out` служить тільки для передачі значення за межі методу. Тому змінній, що використовується як параметр `out`, не потрібно привласнювати якоесь значення. Більше того, в методі параметр `out` вважається неініціалізованим, тобто передбачається, що у нього відсутнє первинне значення. Це означає, що значення має бути привласнене цьому параметру в методі до його завершення. Отже, після виклику методу параметр `out` міститиме деяке значення.

При виклику методу з головного методу `Main()` йому передається як вхідний параметр значення радіуса  $R = 5$ .

**Задача 6.11.** Створити метод, який би обчислював кількість і суму додатних елементів одновимірного масиву дійсних чисел  $A(n)$ .

Блок-схему алгоритму подано на рис.6.14.

*Програмний код:*

```
using System;

namespace C307
{
    class Program
    {
        static void ps(double[] A, out int K, out double s)
        {
            // Метод обчислює кількість і суму додатних
            // елементів одновимірного масиву
            int i;
            K = 0; s = 0;
            for (i = 0; i < A.Length; i++)
            {
                if (A[i] > 0)
                {
                    K = K + 1;
                    s = s + A[i];
                }
            }
        }
    }
}
```

```

static void Main(string[] args)
{
    const int n = 15; // Кількість елементів масиву
    double[] A = new double[n] { 1, 2, -3, 4, -5, 6, -7, 8,
                                  9, -10, 11, -12, 13, -14, 15 };

    int i;
    Console.WriteLine(" Масив A: ");
    for (i = 0; i < n; i++)
    {
        Console.WriteLine($" A[{i}] = {A[i]:F2}");
    }
    Console.WriteLine();
    int K1;
    double s1;
    // Виклик методу ps
    ps(A, out K1, out s1);
    Console.WriteLine($" Кількість додатних елементів = {K1}");
    Console.WriteLine($" Сума додатних елементів = {s1}");
    Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 6.13.



```

D:\Program\C307\C307\bin\Debug\C307.exe
Масив A:
A[0]= 1,00
A[1]= 2,00
A[2]= -3,00
A[3]= 4,00
A[4]= -5,00
A[5]= 6,00
A[6]= -7,00
A[7]= 8,00
A[8]= 9,00
A[9]= -10,00
A[10]= 11,00
A[11]= -12,00
A[12]= 13,00
A[13]= -14,00
A[14]= 15,00

Кількість додатних елементів = 9
Сума додатних елементів = 69

```

Рисунок 6.13 – Результати розв’язання задачі 6.11

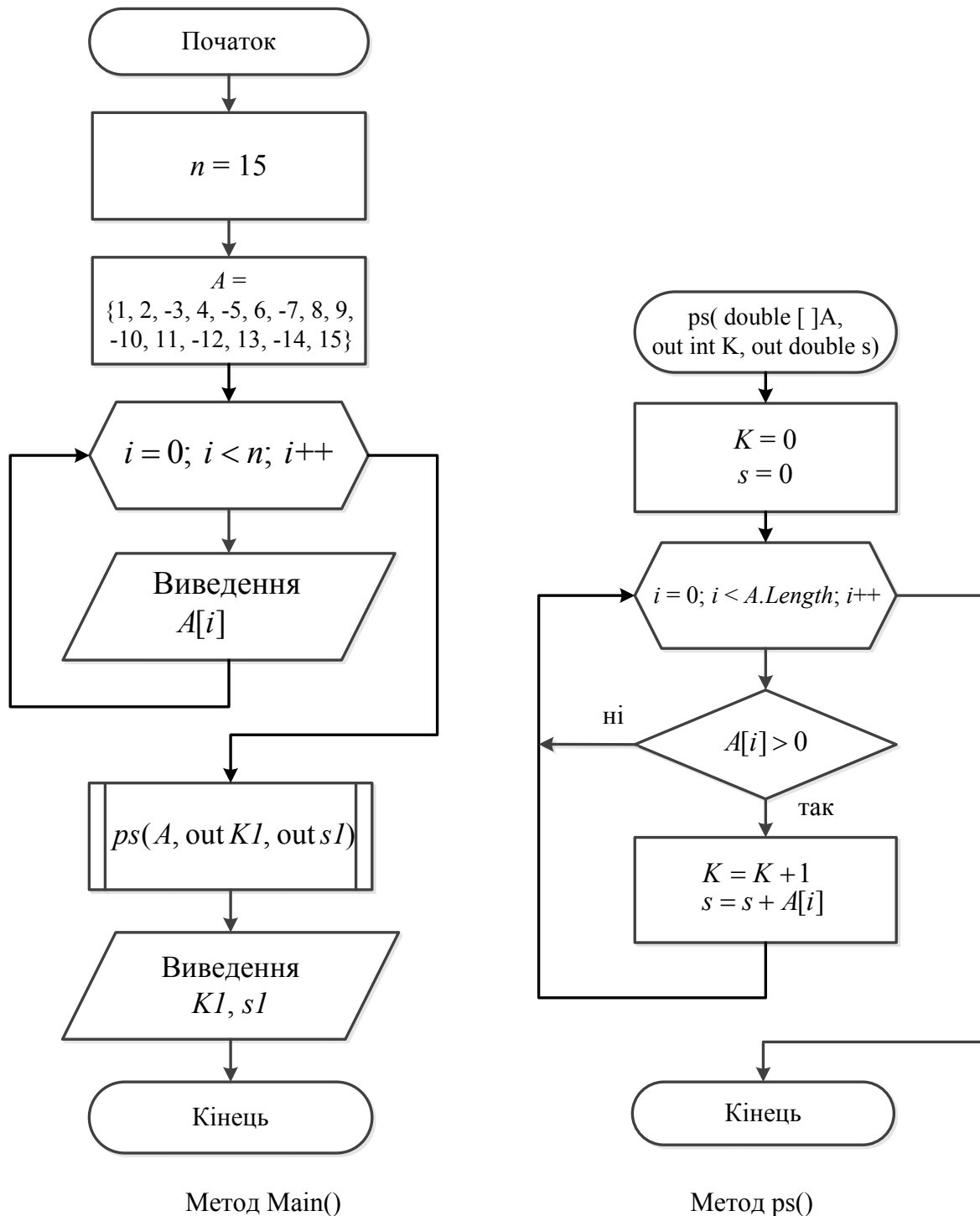


Рисунок 6.14 – Блок-схема алгоритму до задачі 6.11

**Задача 6.12. Обчислити F**

$$F = (K1 + 2K2) \cdot (2M1 - M2) / (imax1 + imax2),$$

де

$K1$  – кількість нульових чисел масиву  $A(n)$ ;

$K2$  – кількість нульових чисел масиву  $B(k)$ ;

$M1$  – максимальний елемент масиву  $A(n)$ ;

$M2$  – максимальний елемент масиву  $B(k)$ ;

$imax1$  – індекс максимального елемента масиву  $A(n)$ ;

$imax2$  – індекс максимального елемента масиву  $B(k)$ .

*Примітка:*

1) значення  $n$ ,  $k$ , спосіб створення масиву, тип елементів вибрати самостійно;

2) для визначення значень  $K1$ ,  $K2$ ,  $M1$ ,  $M2$ ,  $imax1$ ,  $imax2$  розробити методи.

Блок-схему алгоритму подано на рис. 6.16, 6.17.

*Програмний код:*

```
using System;
```

```
namespace C306
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void ps(double[] A, out int KOL)
```

```
        {
```

```
            // Метод обчислює кількість нульових чисел
```

```
            // одновимірного масиву
```

```
            int i;
```

```
            KOL = 0;
```

```
            for (i = 0; i < A.Length; i++)
```

```
            {
```

```
                if (A[i] == 0)
```

```
                {
```

```
                    KOL = KOL + 1;
```

```
                }
```

```
            }
```

```
        }
```

```
        static void MAXIM(double[] A, out double max, out int imax)
```

```
        {
```

```
            // Метод знаходить максимальний елемент масиву
```

```
            // та його індекс
```

```
            int i;
```

```
            max = A[0]; imax = 0;
```

```

    for (i = 1; i < A.Length; i++)
    {
        if (A[i] > max)
        {
            max = A[i];
            imax = i;
        }
    }
}

static void Main(string[] args)
{
    const int n = 10; // Кількість елементів масиву A
    const int k = 12; // Кількість елементів масиву B
    double[] A = new double[n] { 1, -2, 0, 4, 15, 0, -7, 8,
                                  0, 10 };
    double[] B = new double[k] { 1, 0, -3, -4, 5, 6, 7, -8,
                                  19, 0, 11, -12 };

    int i;
    Console.WriteLine(" Масив A: ");
    for (i = 0; i < n; i++)
    {
        Console.WriteLine($" A[{i}]= {A[i]:F2}");
    }
    Console.WriteLine();
    Console.WriteLine(" Масив B: ");
    for (i = 0; i < k; i++)
    {
        Console.WriteLine($" B[{i}]= {B[i]:F2}");
    }
    Console.WriteLine();
    int K1, K2;
    int imax1, imax2;
    double M1, M2, F;
    // Виклик методу ps
    ps(A, out K1);
    ps(B, out K2);
    Console.WriteLine($" Кількість нульових чисел масиву A= {K1}");
}

```

```

    Console.WriteLine($" Кількість нульових чисел масиву B= {K2}");
    // Виклик методу MAXIM
    MAXIM(A, out M1, out imax1);
    MAXIM(B, out M2, out imax2);
    Console.WriteLine();
    Console.WriteLine($" Максимальний елемент масиву A:
                        A[{imax1}] = {M1:F2}");
    Console.WriteLine($" Максимальний елемент масиву B:
                        B[{imax2}] = {M2:F2}");

    Console.WriteLine();
    F = (K1 + 2 * K2) * (2 * M1 - M2) / (imax1 + imax2);
    Console.WriteLine($" F= {F:F3}");
    Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 6.15.

```

D:\Program\C306\C306\bin\Debug\C306.exe
Масив А:
A[0]= 1,00
A[1]= -2,00
A[2]= 0,00
A[3]= 4,00
A[4]= 15,00
A[5]= 0,00
A[6]= -7,00
A[7]= 8,00
A[8]= 0,00
A[9]= 10,00

Масив В:
B[0]= 1,00
B[1]= 0,00
B[2]= -3,00
B[3]= -4,00
B[4]= 5,00
B[5]= 6,00
B[6]= 7,00
B[7]= -8,00
B[8]= 19,00
B[9]= 0,00
B[10]= 11,00
B[11]= -12,00

Кількість нульових чисел масиву А= 3
Кількість нульових чисел масиву В= 2

Максимальний елемент масиву А:  А[4]= 15,00
Максимальний елемент масиву В:  В[8]= 19,00

F= 6,417

```

Рисунок 6.15 – Результати розв’язання задачі 6.12

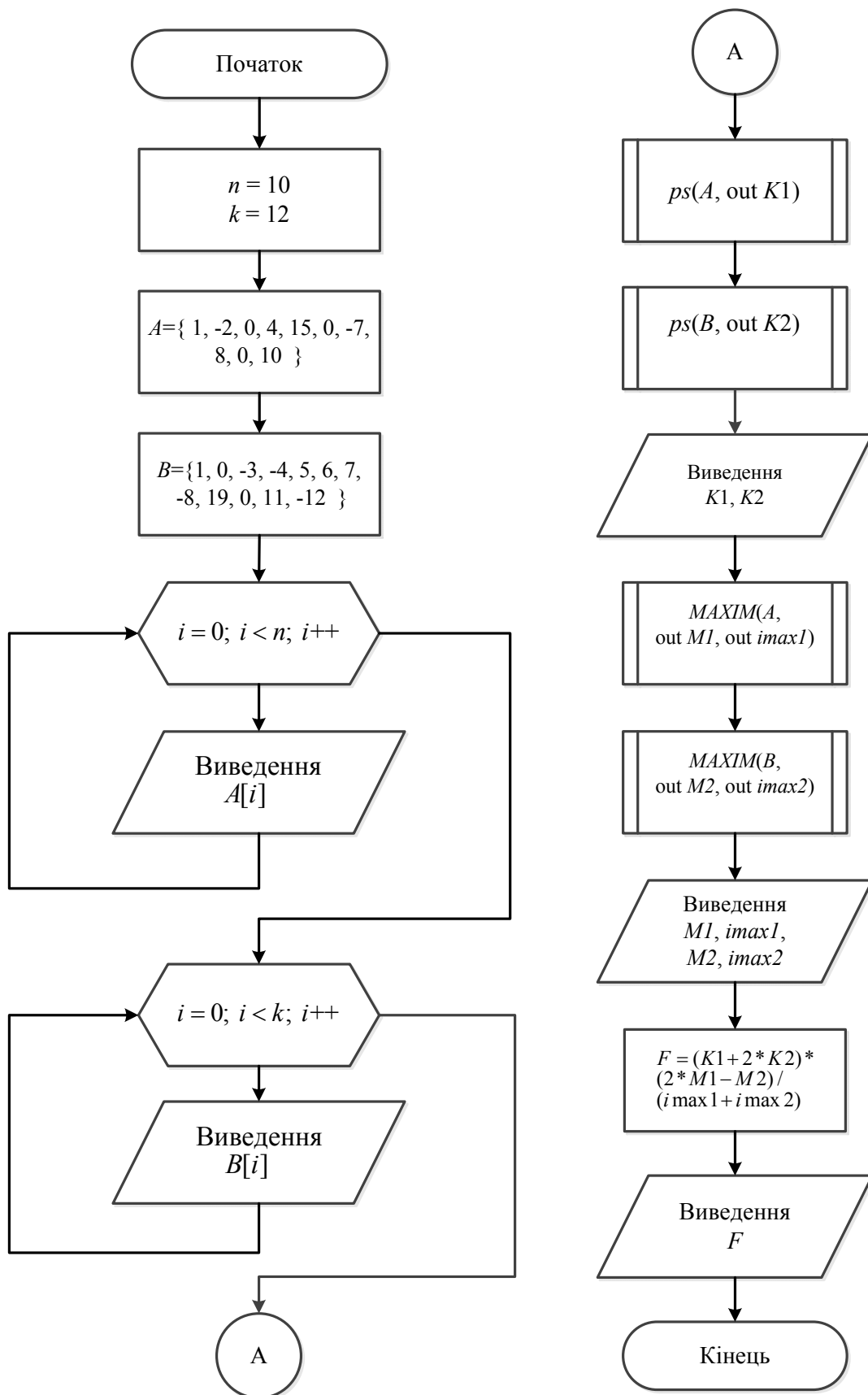


Рисунок 6.16 – Блок-схема алгоритму до задачі 6.12 – Метод Main()

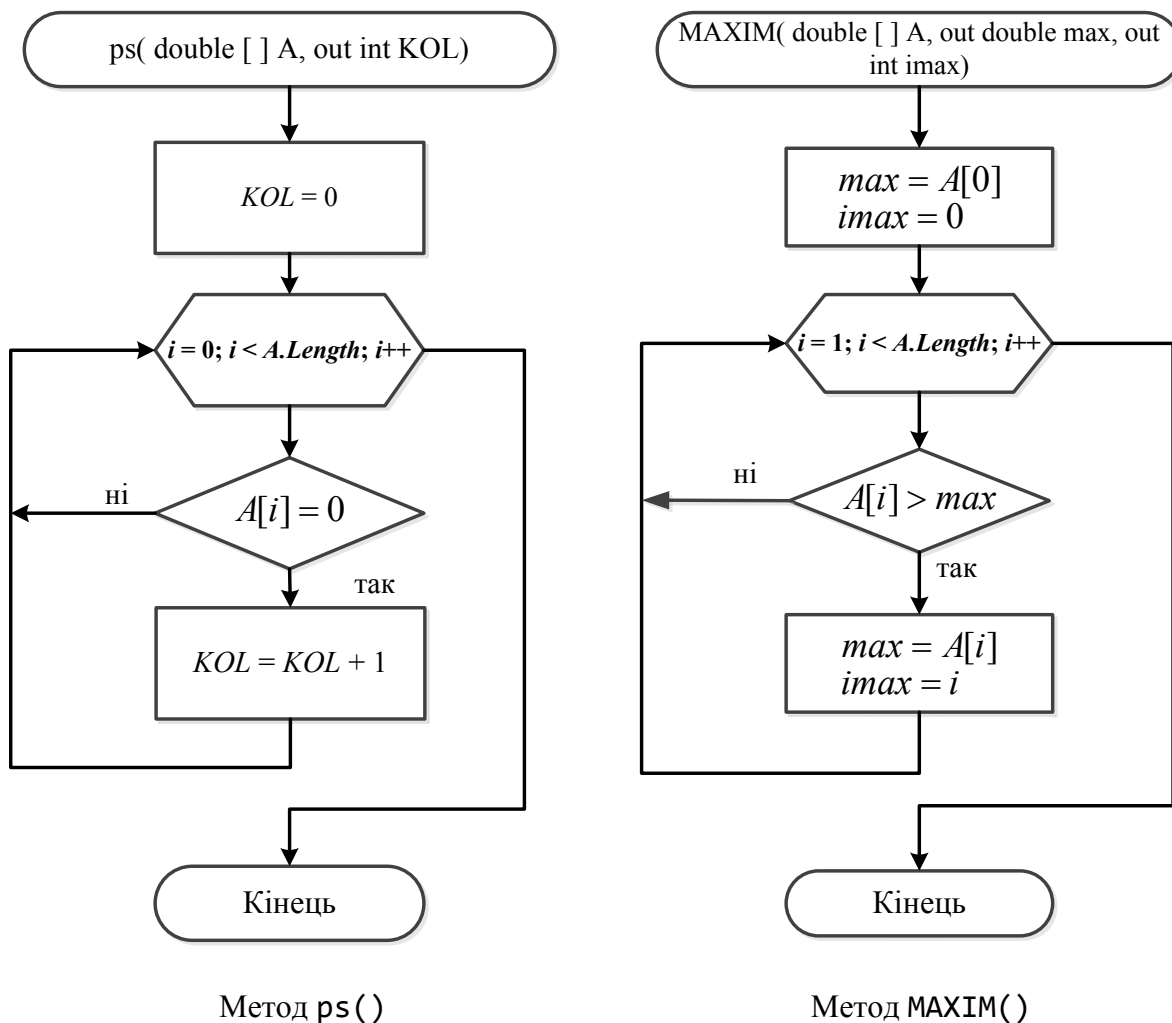


Рисунок 6.17 – Блок-схема алгоритму до задачі 6.12 – метод `ps()` і метод `MAXIM()`

**Задача 6.13.** Написати метод, який знаходить номер першого від’ємного елемента в одновимірному масиві, за відсутності від’ємних чисел повертати повідомлення «Від’ємних чисел немає».

*Програмний код:*

```
using System;

namespace C119
{
    class Program
    {
        static void ps(int[] mas, out int k)
        {
            int i;
            k = -8;
        }
    }
}
```

```

        for (i = 0; i < mas.Length; i++)
            if (mas[i] < 0)
            {
                k = i;
                break;
            }
    }

    static void Main(string[] args)
    {
        int[] M = { 5, 9, 2, 6, -7, 56, 100 };
        int i, p;
        Console.WriteLine(" Масив M ");
        for (i = 0; i < M.Length; i++)
            Console.WriteLine($" M[{i}]= {M[i]}");
        Console.WriteLine();
        ps(M, out p);
        if (p < 0) Console.WriteLine(" Від'ємних чисел немає ");
        else
            Console.WriteLine($" Номер першого від'ємного
                                елемента {p}");

            Console.ReadLine();
    }
}
}

```

Результати розрахунку подано на рис. 6.18.

```

D:\Program\C119\C119\bin\Debug\C119.exe
Масив M
M[0]= 5
M[1]= 9
M[2]= 2
M[3]= 6
M[4]= -7
M[5]= 56
M[6]= 100
Номер першого від'ємного елемента 4

```

Рисунок 6.18 – Результати розв'язання задачі 6.13

**Задача 6.14.** Обчислити значення виразу

$$F = 1,5 \sqrt[3]{\sum_{i=1}^7 (x_i + a)} - 8,2 \sqrt{\sum_{i=0}^{10} (y_i + b)},$$

де  $a = 3; b = 7;$

$x_i$  – елементи одновимірного масиву  $X(12); i = 0, 1, \dots, 11;$

$y_i$  – додатні елементи одновимірного масиву  $Y(12); i = 0, 1, \dots, 11.$

Елементи масивів обчислити за формулами:

$$x_i = i/2, 0 + 5; \quad y_i = i/4, 0 + 1.$$

У вигляді методу оформити обчислення суми  $k1 \cdot k2 \sqrt{\sum_{i=n1}^{n2} (z_i + k3)}$  та розрахування кількості підсумовуваних елементів.

Блок-схему до задачі 6.14 наведено на рис. 6.20.

*Програмний код:*

```
using System;
```

```
namespace C308
```

```
{
```

```
    class Program
```

```
    {
```

```
        static void sum(double k1, double k2, int n1, int n2,  
double[] Z, double k3, out double s, out int q)
```

```
        {
```

```
            int i;
```

```
            s = 0; q = 0;
```

```
            for (i = n1; i <= n2; i++)
```

```
            {
```

```
                s = s + (Z[i] + k3);
```

```
                q = q + 1;
```

```
            }
```

```
            s = k1 * Math.Pow(s, (1 / k2));
```

```
        }
```

```
        static void Main(string[] args)
```

```
        {
```

```
            const int n = 12; // Кількість елементів масиву
```

```
            const double a = 3.0, b = 7.0;
```

```

double[] X = new double[n];
double[] Y = new double[n];
double F, s1, s2;
int i, q1, q2;
Console.WriteLine(" Массив X: ");
for (i = 0; i < n; i++)
{
    X[i] = i / 2.0 + 5;
    Console.Write($"{X[i]:F2} ");
}
Console.WriteLine("\n");
Console.WriteLine(" Массив Y: ");
for (i = 0; i < n; i++)
{
    Y[i] = i / 4.0 + 1;
    Console.Write($"{Y[i]:F2} ");
}
Console.WriteLine("\n");
sum(1.5, 3, 1, 7, X, a, out s1, out q1);
sum(8.2, 2, 0, 10, Y, b, out s2, out q2);
F = s1 - s2;
Console.WriteLine($" F= {F:F3}  q1= {q1}  q2= {q2}");
Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 6.19.

```

D:\Program\C308\C308\bin\Debug\C308.exe
Массив X:
5,00 5,50 6,00 6,50 7,00 7,50 8,00 8,50 9,00 9,50 10,00 10,50
Массив Y:
1,00 1,25 1,50 1,75 2,00 2,25 2,50 2,75 3,00 3,25 3,50 3,75
F= -76,532  q1= 7  q2= 11

```

Рисунок 6.19 – Результати розв’язання задачі 6.14

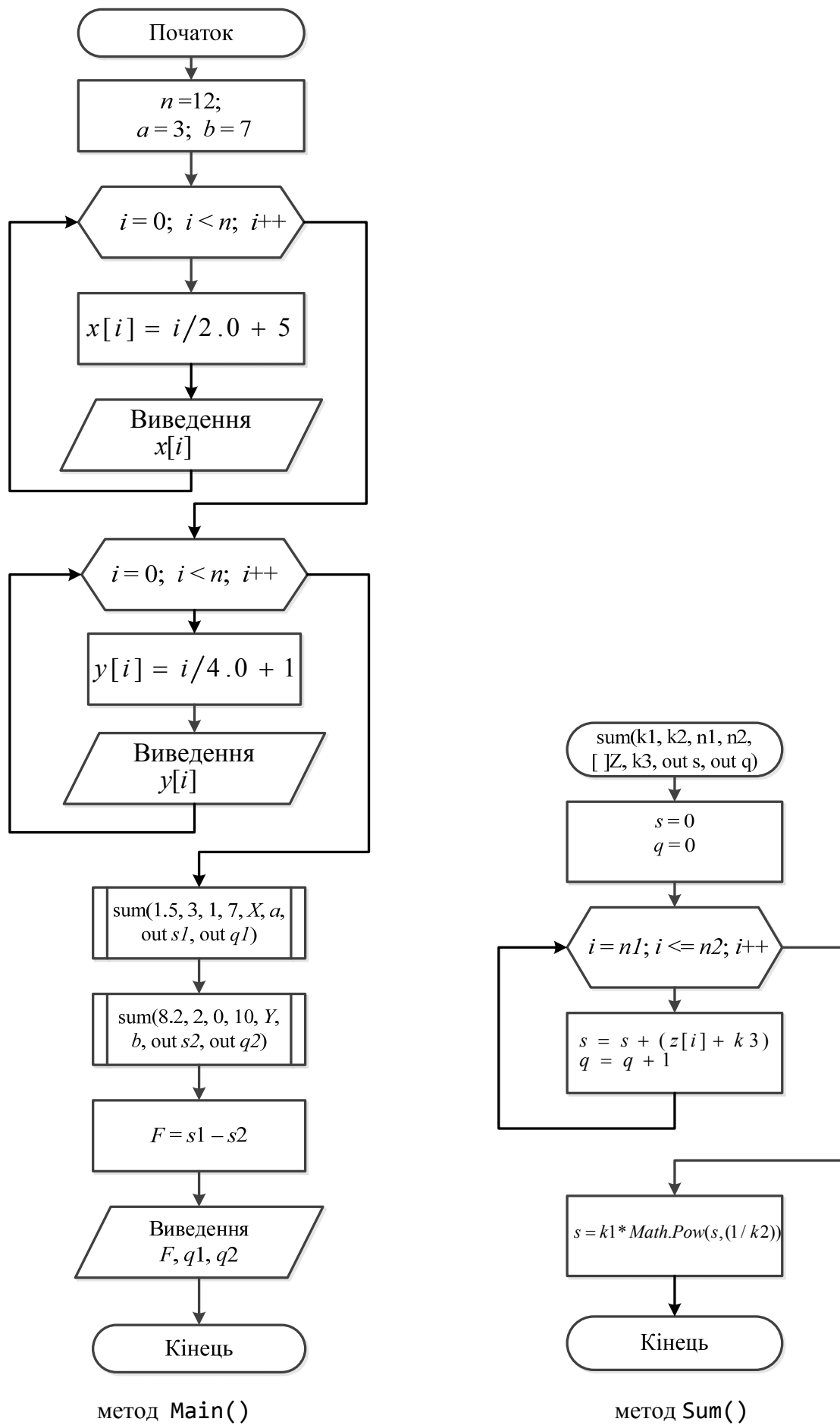


Рисунок 6.20 – Блок-схема алгоритму до задачі 6.14

**Задача 6.15.** Обчислити значення виразу  $z$

$$z = 0,5 \prod_{i=3}^{12} (b_i - 3)^2 + 0,2 \prod_{i=0}^{14} (b_i + 0,82),$$

де  $b_i$  – елементи одновимірного масиву  $B(15)$ ;  $i = 0, 1, \dots, 14$ .

У вигляді методу оформити обчислення добутку виду  $k1 \prod_{i=n1}^{n2} (b_i + k2)^{k3}$  та кількість елементів, що перемножуються.

Блок-схему наведено на рис. 6.22, 6.23.

*Програмний код:*

```
using System;

namespace C61a
{
    class Program
    {
        static void ps(int n1, int n2, double k1, double k2, double
k3, double[] B, out double p, out int q)
        {
            int i;
            p = 1; q = 0;
            for (i = n1; i <= n2; i++)
            {
                p = p * Math.Pow((B[i] + k2), k3);
                q = q + 1;
            }
            p = k1 * p;
        }

        static void Main(string[] args)
        {
            const int n = 15; // Кількість елементів масиву
            double[] B = new double[n] { 1.0, 2.0, 3.0, 4.0, 5.0,
6.0, 7.0, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 14.0, 15.0};
            double z, p1, p2;
            int q1, q2;
            Console.WriteLine(" Масив B: ");
            for (int i = 0; i < n; i++)
                Console.WriteLine($" B[{i}]= {B[i]:F2}");
            Console.WriteLine();
            ps(3, 12, 0.5, -3, 2, B, out p1, out q1);
            ps(0, 14, 0.2, 0.82, 1, B, out p2, out q2);
            z = p1 + p2;
            Console.WriteLine($" z= {z}   q1= {q1}   q2= {q2}");
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 6.21.

```
D:\Program\C61a\C61a\bin\Debug\C61a.exe
Масив В:
В[0]= 1,00
В[1]= 2,00
В[2]= 3,00
В[3]= 4,00
В[4]= 5,00
В[5]= 6,00
В[6]= 7,00
В[7]= 8,00
В[8]= 9,00
В[9]= 10,00
В[10]= 11,00
В[11]= 12,00
В[12]= 13,00
В[13]= 14,00
В[14]= 15,00
z = 9283391065680,24  q1 = 10  q2 = 15
```

Рисунок 6.21 – Результати розв’язання задачі 6.15

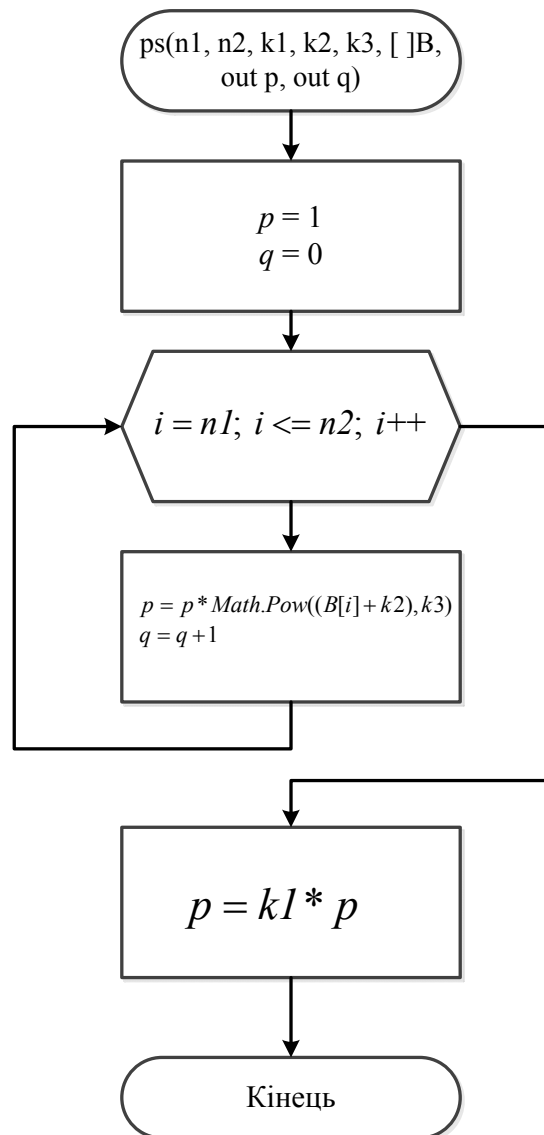


Рисунок 6.22 – Блок-схема алгоритму до задачі 6.15 – метод ps()

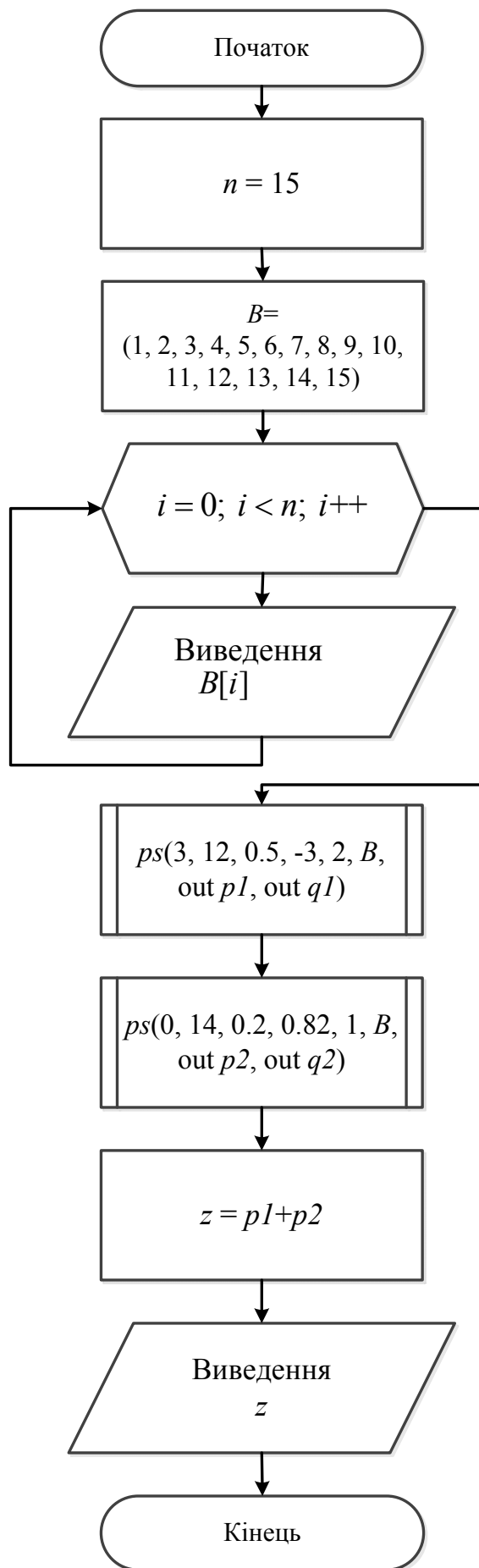


Рисунок 6.23 – Блок-схема алгоритму до задачі 6.15 – метод Main()

**Задача 6.16.** Написати метод, що забезпечує розв'язання квадратного рівняння.

*Програмний код:*

```
using System;

namespace C317
{
    class Program
    {
        // Метод розв'язання квадратного рівняння
        static void kvadur(double a, double b, double c,
            ref double x1, ref double x2, ref Boolean pr)
        {
            double D;
            D = b * b - 4 * a * c;
            if (D >= 0 && a != 0)
            {
                pr = true;
                x1 = (-b + Math.Sqrt(D)) / (2 * a);
                x2 = (-b - Math.Sqrt(D)) / (2 * a);
            }
            else
            {
                pr = false;
            }
        }

        static void Main(string[] args)
        {
            double a, b, c, x1, x2;
            Boolean pr;
            Console.WriteLine(" Введіть a, b, c ? ");
            a = Convert.ToDouble(Console.ReadLine());
            b = Convert.ToDouble(Console.ReadLine());
            c = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine();
            // значення аргументам необхідно присвоїти до виклику методу
            x1 = 1; x2 = 1; pr = true;
            kvadur(a, b, c, ref x1, ref x2, ref pr);
            if (pr)
                Console.WriteLine($" x1= {x1:F3}  x2= {x2:F3}");
        }
    }
}
```

```

else
    Console.WriteLine(" Рівняння не має розв'язку");
Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 6.24.

```

D:\Program\C317\C317\bin\Debug\C317.exe
Введіть а, б, с ?
3
8
2
x1= -0,279 x2= -2,387

```

```

D:\Program\C317\C317\bin\Debug\C317.exe
Введіть а, б, с ?
12
10
20
Рівняння не має розв'язку

```

Рисунок 6.24 – Результати розв'язання задачі 6.16

**Задача 6.17.** У мові програмування C# результатом методу може бути й масив, що ілюструє наступний приклад формування нового масиву з додатних елементів початкового масиву.

*Програмний код:*

```

using System;

namespace C146
{
    class Program
    {
        static int[] fun1(int[] k)
        { // результат функції - масив
            int koL = 0;
            int j;
            for (int i = 0; i < k.Length; i++)
                if (k[i] > 0) koL = koL + 1;
            int[] res = new int[koL]; // оголошення результату-масиву
            j = 0;

```

```

    for (int i = 0; i < k.Length; i++)
    {
        if (k[i] > 0)
        {
            res[j] = k[i];
            j = j + 1;
        }
    }
    return res; // повернення масиву як результату
}

static void Main(string[] args)
{
    int[] mas; // початковий масив
    int[] arr; // масив-результат
    int n;
    // Введення елементів масиву
    Console.Write(" Введіть кількість елементів = ? ");
    n = Convert.ToInt32(Console.ReadLine());
    mas = new int[n];
    for (int i = 0; i < mas.Length; i++)
    {
        Console.Write($" mas[{i}]= ? ");
        mas[i] = Convert.ToInt32(Console.ReadLine());
    }
    Console.WriteLine();
    // Виведення елементів початкового масиву
    Console.WriteLine("Початковий масив: ");
    for (int i = 0; i < mas.Length; i++)
    {
        Console.WriteLine($" mas[{i}]= {mas[i]}");
    }
    arr = fun1(mas); // ініціалізація масиву-результату не потрібна
    // Виведення масиву-результату
    Console.WriteLine("\n масив-результат:");
    for (int i = 0; i < arr.Length; i++)
        Console.WriteLine($" arr[{i}]= {arr[i]}");
    Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 6.25.

```

D:\Program\C146\C146\bin\Debug\C146.exe
Введіть кількість елементів = ? ?
mas[0]= ? 1
mas[1]= ? -2
mas[2]= ? 3
mas[3]= ? -5
mas[4]= ? -7
mas[5]= ? 9
mas[6]= ? 12

Початковий масив:
mas[0]= 1
mas[1]= -2
mas[2]= 3
mas[3]= -5
mas[4]= -7
mas[5]= 9
mas[6]= 12

масив-результат:
arr[0]= 1
arr[1]= 3
arr[2]= 9
arr[3]= 12

```

Рисунок 6.25 – Результати розв’язання задачі 6.17

**Задача 6.18.** Створити проект, в якому для виконання повторюваних дій використовуються методи:

а) організувати двовимірний масив  $A$  розміром  $n \times m$ ,

$$A = (a_{ij}; i = 0, 1, \dots, n-1; j = 0, 1, \dots, m-1),$$

елементи масиву задати в програмному кодї;

б) отримати два одновимірних масиви:  $SS1(m)$  – сума елементів стовпців;  $PS1(n)$  – добуток елементів рядків. Вивести масиви  $SS1(m)$  і  $PS1(n)$  на екран;

в) знайти максимальні елементи масивів  $SS1(m)$  і  $PS1(n)$ , індекси максимальних елементів, різницю між максимальними елементами масивів  $SS1(m)$  і  $PS1(n)$ ;

г) відсортувати масиви  $SS1(m)$  і  $PS1(n)$  за зростанням.

Знаходження суми, добутоків елементів масиву, максимального елемента масиву і сортування елементів масиву здійснити за допомогою методів.

*Програмний код:*

```

using System;

namespace C67
{
    class Program
    {

```

```

// Метод для обчислення суми елементів стовпців матриці
static void ss(double[,] B, int j, out double s)
{
    int i;
    s = 0;
    for (i = 0; i < B.GetLength(0); i++)
        s = s + B[i, j];
}

// Метод для обчислення добутку елементів рядків матриці
static void ps(double[,] B, int i, out double p)
{
    int j;
    p = 1;
    for (j = 0; j < B.GetLength(1); j++)
        p = p * B[i, j];
}

// Метод для знаходження максимального елемента
// одновимірного масиву та його індексу
static void MAXIM(double[] B, out double max, out int index)
{
    int i;
    max = B[0]; index = 0;
    for (i = 1; i < B.Length; i++)
    {
        if (max < B[i])
        {
            max = B[i];
            index = i;
        }
    }
}

static void Main(string[] args)
{
    const int n = 4; // Кількість рядків масиву A
    const int m = 5; // Кількість стовпців масиву A
    double[] SS1 = new double[m];
    double[] PS1 = new double[n];
    double smax, pmax, r;
    int i, j, indexss, indexps;
}

```

```

// Заповнення елементів масиву
double[,] A = new double[n, m]
    {{1.00, 5.00, -3.00, 2.00, 4.00},
     {6.00, -10.00, 7.00, 8.00, 9.00},
     {10.00, 9.00, 6.00, 7.00, 8.00},
     {-4.00, 3.00, 1.00, -2.00, 5.00}};
// Виведення елементів масиву
Console.WriteLine(" Масив A: ");
for (i = 0; i < n; i++)
{
    for (j = 0; j < m; j++)
    {
        Console.Write($"{A[i,j],9:F2} ");
    }
    Console.WriteLine();
}
Console.WriteLine("\n");
// Звернення до методу для отримання сум
// елементів стовпців
Console.WriteLine(" Масив SS1: Сума елементів стовпців ");
for (j = 0; j < m; j++)
{
    ss(A, j, out SS1[j]);
    Console.Write($"{SS1[j],10:F2} ");
}
Console.WriteLine("\n");
// Звернення до методу для отримання добутку
// елементів рядків
Console.WriteLine(" Масив PS1: добуток елементів рядків ");
for (i = 0; i < n; i++)
{
    ps(A, i, out PS1[i]);
    Console.Write($"{PS1[i],10:F2} ");
}
Console.WriteLine("\n");
// Звернення до методу для знаходження максимальних
// елементів масивів і їх індексів
MAXIM(SS1, out smax, out indexss);
Console.WriteLine(" Максимальний елемент в масиві SS1: ");
Console.WriteLine($" SS1[{indexss}] = {smax:F2} ");
Console.WriteLine();
MAXIM(PS1, out pmax, out indexps);
Console.WriteLine($" Максимальний елемент в масиві PS1: ");

```

```

Console.WriteLine($" PS1[{indexps}]= {pmax:F2} ");
Console.WriteLine();
// Визначення різниці між максимальними елементами
r = smax - pmax;
Console.WriteLine($" Різниця між максимальними
                    елементами r= {r:F2}");

Console.WriteLine();
// Сортювання масиву SS1
Console.WriteLine(" Відсортований масив SS1");
Array.Sort(SS1);
// Виведення елементів відсортованого масиву SS1
for (i = 0; i < n; i++)
    Console.Write($"{SS1[i],10:F2} ");
Console.WriteLine("\n");
// Сортювання масиву PS1
Console.WriteLine(" Відсортований масив в PS1");
Array.Sort(PS1);
// Виведення елементів відсортованого масиву PS1
for (i = 0; i < n; i++)
    Console.Write($"{PS1[i],10:F2} ");
Console.WriteLine();
Console.ReadLine();
    }
}
}

```

Результати розрахунку подано на рис. 6.26.

```

D:\Program\C67\C67\bin\Debug\C67.exe
Масив A:
  1,00    5,00   -3,00    2,00    4,00
  6,00   -10,00  7,00    8,00    9,00
 10,00    9,00    6,00    7,00    8,00
  -4,00    3,00    1,00   -2,00    5,00

Масив SS1: Сума елементів стовпців
 13,00    7,00   11,00   15,00   26,00

Масив PS1: добуток елементів рядків
-120,00 -30240,00 30240,00 120,00

Максимальний елемент в масиві SS1:
SS1[4]= 26,00

Максимальний елемент в масиві PS1:
PS1[2]= 30240,00

Різниця між максимальними елементами r= -30214,00

Відсортований масив SS1
 7,00   11,00   13,00   15,00

Відсортований масив в PS1
-30240,00 -120,00 120,00 30240,00

```

Рисунок 6.26 – Результати розв’язання задачі 6.18

*Пояснення:*

1) for(i=1; i < B.Length; i++)

Властивість Length (клас Array) – кількість елементів масиву.

Властивість Length дозволяє реалізовувати алгоритми, які працюватимуть з масивами різної довжини.

2) for (i = 0; i < B.GetLength(0) ; i++)

Метод GetLength(0) повертає кількість рядків двовимірного масиву

3) for (j = 0; j < B.GetLength(1) ; j++)

Метод GetLength(1) повертає кількість стовпців двовимірного масиву

**Задача 6.19** – Методи зі змінною кількістю аргументів

Написати метод обчислення середнього значення декількох чисел.

*Програмний код:*

```
using System;
```

```
namespace C118
```

```
{
```

```
    class Program
```

```
    {
```

```
        public static double ps(params double[] A)
```

```
        {
```

```
            double sum = 0;
```

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

```
                sum = sum + A[i];
```

```
            return sum / A.Length;
```

```
        }
```

```
        static void Main(string[] args)
```

```
        {
```

```
            double p, z, k, s1, s2, s3;
```

```
            p = 15.8; z = 4.5; k = 8.0;
```

```
            double[] b = new double[4] { 10.0, 20.5, 30.2, 38.0 };
```

```
            s1 = ps(p, z, k);
```

```
            s2 = ps(5.2, 15.7, 24.8, 27.0, 32.5);
```

```
            s3 = ps(b);
```

```
            Console.WriteLine($"s1= {s1:F3}  s2= {s2:F3}
```

```
                                s3= {s3:F3}");
```

```
            Console.ReadLine();
```

```
        }
```

```
    }
```

```
}
```

Результати розрахунку подано на рис. 6.27.



Рисунок 6.26 – Результати розв’язання задачі 6.19

*Пояснення:* Іноді вимагається створити метод, в який можна передавати різну кількість аргументів. Мова С# надає таку можливість за допомогою ключового слова `params`. Параметр із ключовим словом `params` розміщується в списку параметрів останнім і означає масив заданого типу невизначеної довжини. Кількість елементів масиву параметрів дорівнюватиме кількості аргументів, що передаються методу. Кількість елементів масиву отримують за допомогою його властивості `Length`.

## 6.2 Завдання для виконання лабораторних робіт

**Варіант 1.** Написати метод, який обчислює площу поверхні та об'єм циліндра  $s = 2\pi r(h + r)$ ,  $v = \pi r^2 h$ .

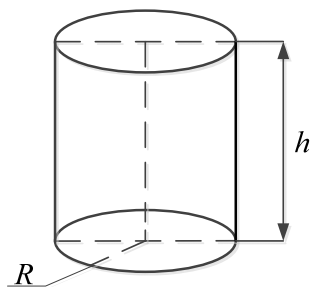


Рисунок 7.27 – Циліндр до задачі варіанта 1

**Варіант 2.** Декілька трикутників задані своїми сторонами. Вивести площі цих трикутників. Знаходження площі трикутника за формулою Герона оформити як метод:

$$S = \sqrt{pp(pp - x)(pp - y)(pp - z)};$$

$$p = x + y + z;$$

$$pp = p/2,$$

де  $x, y, z$  – сторони трикутника (вхідні параметри);  
 $p$  і  $s$  – периметр і площа (вихідні параметри).

**Варіант 3.** Напишіть метод, який мінятиме в масиві цілих чисел усі елементи, які дорівнюють вказаному значенню на протилежне значення за знаком. Наприклад, усі елементи масиву, які дорівнюють 5, мінятимуться на  $-5$ .

**Варіант 4.** Створити метод, який виводить піднесені до квадрата значення елементів одновимірного масиву  $C(p)$  на екран.

**Варіант 5.** Створити метод, який підраховує кількість парних чисел у масиві цілих чисел  $A(n)$  і замінює їх на  $-10$ .

**Варіант 6.** Обчислити F

$$F = N! + (N - M)! - (2N)!,$$

де  $N!$  – факторіал числа  $N$ .

Обчислення факторіала оформити у вигляді методу.

Початкові дані:  $N = 12$ ;  $M = 7$ .

**Варіант 7.** Обчислити  $F$

$$F = (2K1 - K2) \cdot (S1 - 3S2),$$

де  $K1$  – кількість додатних чисел масиву  $A(n)$ ;

$K2$  – кількість додатних чисел масиву  $B(k)$ ;

$S1$  – сума додатних чисел масиву  $A(n)$ ;

$S2$  – сума додатних чисел масиву  $B(k)$ .

*Примітка:*

1) значення  $n$ ,  $k$ , спосіб створення масиву, тип елементів вибрати самостійно;

2) для визначення значень  $K1$ ,  $K2$ ,  $S1$ ,  $S2$  розробити методи.

**Варіант 8.** Обчислити  $F$

$$F = (3K1 - 4K2) / (2P1 - P2),$$

де  $K1$  – кількість від'ємних чисел масиву  $A(n)$ ;

$K2$  – кількість від'ємних чисел масиву  $B(k)$ ;

$P1$  – добуток додатних чисел масиву  $A(n)$ ;

$P2$  – добуток додатних чисел масиву  $B(k)$ .

*Примітка:*

1) значення  $n$ ,  $k$ , спосіб створення масиву, тип елементів вибрати самостійно;

2) для визначення значень  $K1$ ,  $K2$ ,  $P1$ ,  $P2$  розробити методи.

**Варіант 9.** Створити метод, що приймає як аргументи масив дійсних чисел  $B(k)$  і повертає суму чисел масиву, піднесену до квадрата як результат.

**Варіант 10.** Обчислити  $F$

$$F = (P1 - 3P2) / (2M1 + 3M2),$$

де  $P1$  – добуток від'ємних чисел  $A(n)$ ;

$P2$  – добуток від'ємних чисел  $B(k)$ ;

$M1$  – максимальний елемент масиву  $A(n)$ ;

$M2$  – максимальний елемент масиву  $B(k)$ .

*Примітка:*

- 1) значення  $n$ ,  $k$ , спосіб створення масиву, тип елементів вибрати самостійно;
- 2) для визначення значень  $P1$ ,  $P2$ ,  $M1$ ,  $M2$  розробити методи.

**Варіант 11.** Створити метод, який підраховує кількість непарних чисел у масиві цілих чисел  $B(k)$  і замінює їх на  $-1$ .

**Варіант 12.** Створити метод, який би в одновимірному масиві  $A(n)$  усі від'ємні елементи підносив би до квадрата.

**Варіант 13.** Створити метод, який би обчислював середнє арифметичне додатних елементів одновимірного масиву дійсних чисел  $A(n)$ .

**Варіант 14.** Створити метод, який би в одновимірному масиві  $A(n)$  усі нульові елементи заміняв на 10.

**Варіант 15.** Створити метод, який би в одновимірному масиві  $A(n)$  обчислював кількість елементів, рівних нулю, і визначав їх індекси.

**Варіант 16.** Дано три одновимірних масиви цілих чисел  $A(n)$ ,  $B(m)$ ,  $C(p)$ . Знайти суму елементів, розташованих до першого від'ємного елемента (хоч один від'ємний елемент є присутнім завжди). Використати метод.

**Варіант 17.** Створити метод, який би в одновимірному масиві  $A(n)$  усі додатні елементи підносив би до квадрата.

**Варіант 18.** Створити метод, який би в одновимірному масиві  $A(n)$  визначав суму елементів, кратних трьом.

**Варіант 19.** Дано три одновимірних масиви цілих чисел  $A(n)$ ,  $B(m)$ ,  $C(p)$ . Знайти суму елементів, розташованих після першого нульового елемента (хоча один нульовий елемент є присутнім завжди). Використати метод.

**Варіант 20.** Написати метод для розрахування середнього арифметичного значення елементів двовимірного масиву.

## Тема 7. РЯДКИ

*Мета роботи:* вивчити методи роботи з рядками.

### 7.1. Приклади розв'язання задач

#### Задача 7.1. Операції з рядками

*Програмний код:*

```
using System;

namespace C77a
{
    class Program
    {
        static void Main(string[] args)
        {
            string s1 = "Типи даних у мові програмування C#";
            string s2 = "C#";
            string s3 = string.Copy(s1);
            string s4 = new string('-', 70);

            // Виведення рядка s1
            Console.WriteLine("Рядок:\t\t" + s1);

            // Виведення довжини рядка
            Console.WriteLine("Довжина рядка:\t" + s1.Length);

            //Виведення рядка в нижньому регістрі - усі символи малі
            Console.WriteLine("Усі малі:\t" + s1.ToLower());

            //Виведення рядка у верхньому регістрі - усі великі
            Console.WriteLine("Усі великі:\t" + s1.ToUpper());

            // Порівняння двох рядків
            Console.WriteLine("Порівняння:\t" + String.Equals(s1, s2));

            // Рядок s1 містить підрядок s2?
            Console.WriteLine("s1 містить s2? " + s1.Contains(s2));

            //З якої позиції рядок s2 входить у рядок s1
            Console.WriteLine("s2 вх. у s1 з п." + s1.IndexOf(s2));

            //Усі входження рядка s2 в рядку s1 замінити рядком "C++"
            Console.WriteLine("Заміна:\t\t" + s1.Replace(s2, "C++"));
        }
    }
}
```

```

//Вилучається підрядок завдовжки 13 символів, починаючи з 18
Console.WriteLine("Підрядок:\t" + s1.Substring(18, 13));

// Видаляє 19 символів, починаючи з 13-го
Console.WriteLine("Видалення:\t" + s1.Remove(13, 19));

//У рядок s1 вставляє рядок "високого рівня" за індексом 32
Console.WriteLine("Вставка:\t" + s1.Insert(32, "високого
                                     рівня "));

Console.WriteLine(s4);
Console.ReadLine();
    }
}
}

```

Результати розрахунку подано на рис. 7.1.

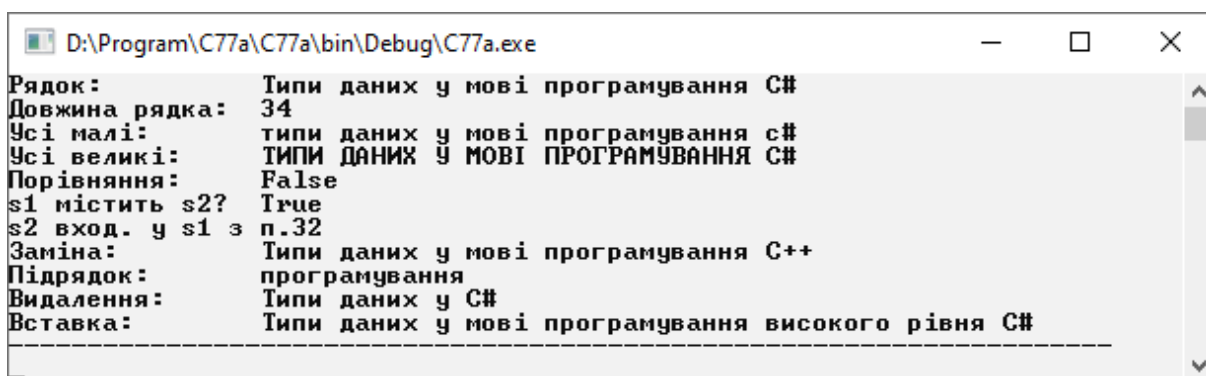


Рисунок 7.1 – Результати розв’язання задачі 7.1

*Пояснення:* рядки `s1` і `s2` ініціалізувалися за допомогою рядкових констант, рядкова змінна `s3` утворена копією `s1`, але є окремим екземпляром. Рядок `s4` утворений набором символів. У програмі проводяться операції з рядками.

**Задача 7.2.** Надрукувати рядок, окремо кожен символ.

*Програмний код:*

```

using System;

namespace C73
{
    class Program
    {
        static void Main(string[] args)
        {
            string str = "Інформатика";

```

```

        Console.WriteLine(" " + str);
        Console.WriteLine();
        for (int i = 0; i < str.Length; i++)
        {
            char ch = str[i];
            Console.WriteLine(" " + ch);
        }
        Console.ReadLine();
    }
}
}
}

```

Результати розрахунку подано на рис. 7.2.



Рисунок 7.2 – Результати розв’язання задачі 7.2

### Задача 7.3 – Пошук кількості слів у реченні

Написати програму, в якій увести рядок з клавіатури, обчислити кількість слів, розділених пробілом (як розділювач допускаються тільки одиничні пробіли). Результат вивести на консоль.

*Програмний код:*

```

using System;

namespace C80
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Введіть текст: ");
            string s1 = Console.ReadLine();
            int word = 0;
            int i;

```

```

        for (i = 0; i < s1.Length; i++)
            if (s1[i] == ' ')
                word = word + 1;
        Console.WriteLine("Кількість слів: " + (word + 1));
        Console.ReadLine();
    }
}

```

Результати розрахунку подано на рис. 7.3.

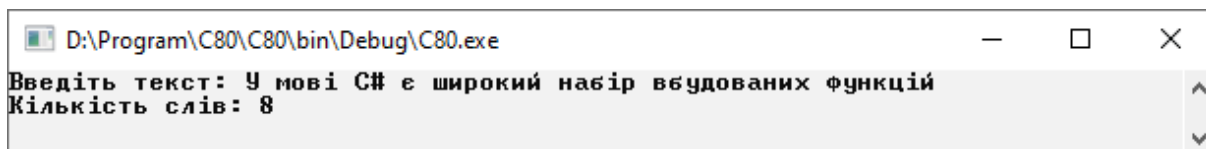


Рисунок 7.3 – Результати розв’язання задачі 7.3

**Задача 7.4.** У цьому тексті замінити усі літери «а» на «о», а літери «м» на «н».

### *Спосіб 1*

*Програмний код:*

```

using System;

namespace C88
{
    class Program
    {
        static void Main(string[] args)
        {
            string a = "інформатика";
            string b;
            Console.WriteLine("Початковий текст: " + a);
            b = a.Replace('a', 'o');
            b = b.Replace('m', 'n');
            Console.WriteLine("Новий текст: " + b);
            Console.ReadLine();
        }
    }
}

```

Результати розрахунку подано на рис. 7.4.

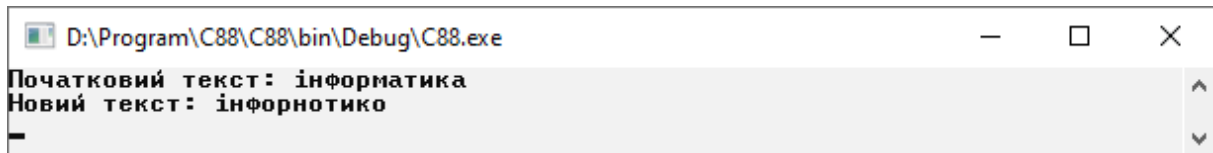


Рисунок 7.4 – Результати розв’язання задачі 7.4 (спосіб 1)

### *Спосіб 2*

*Програмний код:*

```
using System;

namespace C85
{
    class Program
    {
        static void Main(string[] args)
        {
            string a = "інформатика", b = "";
            char c;
            Console.WriteLine("Початковий текст: " + a);
            for (int i = 0; i < a.Length; i++)
            {
                c = a[i];
                if (c == 'a')
                    b = b + 'o';
                else if (c == 'м')
                    b = b + 'н';
                else
                    b = b + c;
            }
            Console.WriteLine("Новий текст: " + b);
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 7.5.



Рисунок 7.5 – Результати розв’язання задачі 7.4 (спосіб 2)

### Спосіб 3

*Програмний код:*

```
using System;

namespace C87
{
    class Program
    {
        static void Main(string[] args)
        {
            string a = "інформатика", b = "";
            string c;
            Console.WriteLine("Початковий текст: " + a);
            for (int i = 0; i < a.Length; i++)
            {
                c = a.Substring(i, 1);
                if (c == "а")
                    b = b + "о";
                else if (c == "м")
                    b = b + "н";
                else
                    b = b + c;
            }
            Console.WriteLine("Новий текст: " + b);
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 7.6.



Рисунок 7.6 – Результати розв’язання задачі 7.4 (спосіб 3)

### Задача 7.5 – Посимвольний перебір

У заданому рядку підрахувати кількість голосних букв.

*Програмний код:*

```
using System;

namespace C108
{
    class Program
    {
```



### Задача 7.6 – Реверс рядка

Написати програму, що перетворює заданий рядок у зворотному порядку (перший символ стає останнім, останній – першим і так далі). Результат вивести на консоль.

#### Спосіб 1

*Програмний код:*

```
using System;

namespace C79a
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Введіть слово: ");
            string s1 = Console.ReadLine();
            int i;
            for (i = s1.Length - 1; i >= 0; i--)
            {
                Console.Write(s1[i]);
            }
            Console.WriteLine();
            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 7.8.

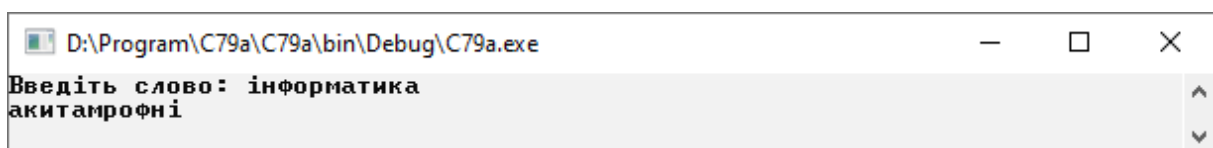


Рисунок 7.8 – Результати розв’язання задачі 7.6 (спосіб 1)

#### Спосіб 2

*Програмний код:*

```
using System;

namespace C79
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Введіть слово: ");

```

```

        string s1 = Console.ReadLine();
        char[] s2 = s1.ToCharArray();
        Array.Reverse(s2);
        s1 = new string(s2);
        Console.WriteLine(s2);
        Console.ReadLine();
    }
}

```

Результати розрахунку подано на рис. 7.9.

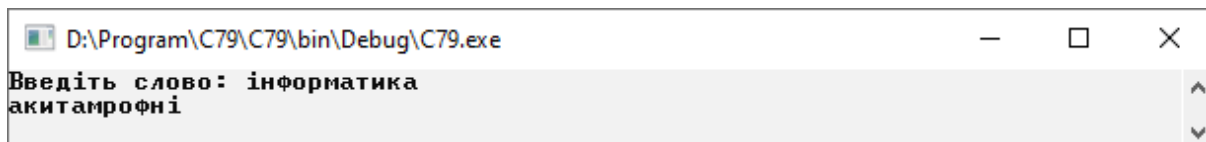


Рисунок 7.9 – Результати розв’язання задачі 7.6 (спосіб 2)

**Задача 7.7.** Увести прізвище, ім’я і по батькові як одне дане рядкового типу. Визначити довжину рядка і кількість літер «А» в ній. Вивести окремо кожне слово. Визначити довжину імені. Вивести першу літеру по батькові.

*Програмний код:*

```

using System;

namespace C92
{
    class Program
    {
        static void Main(string[] args)
        {
            string str = "АСТАХОВА ГАННА ВОЛОДИМИРІВНА";
            string f1; // Прізвище
            string f2; // Ім’я
            string f3; // По батькові
            int i;
            int k; // Довжина рядка
            int k1; // Довжина імені
            int n; // Кількість літер <A> в рядку
            string[] strWords;
            k = str.Length;
            Console.WriteLine("Довжина рядка = " + k + "\n ");
            // Визначаємо кількість літер <A> в рядку

```

```

n = 0;
for (i = 0; i < k; i++)
{
    if ((str[i] == 'A') || (str[i] == 'a'))
        n = n + 1;
}
Console.WriteLine("У рядку кількість літер <A> дорівнює
                    " + n + "\n");
// Рядок розділяємо на складові частини.
// У результаті отримуємо масив, що містить підрядки
strWords = str.Split(" ".ToCharArray());
f1 = strWords[0]; // Прізвище
f2 = strWords[1]; // Ім'я
f3 = strWords[2]; // По батькові
Console.WriteLine(f1 + "\n" + f2 + "\n" + f3 + "\n");
// Визначаємо довжину імені
k1 = f2.Length;
Console.WriteLine("Довжина імені = " + k1);
// Виводимо першу букву по батькові
Console.WriteLine(f3[0]);
Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 7.10.

```

D:\Program\C92\C92\bin\Debug\C92.exe
Довжина рядка = 28
У рядку кількість літер <A> дорівнює 6
АСТАХОВА
ГАННА
ВОЛОДИМИРІВНА
Довжина імені = 5
В

```

Рисунок 7.10 – Результати розв’язання задачі 7.7

## 7.2 Завдання для виконання лабораторних робіт

### ВАРІАНТ 1

Увести прізвище, ім’я і по батькові як одне дане рядкового типу  
 " ПАВЛОВСЬКА ІРИНА СТЕПАНІВНА "

Визначити довжину рядка і кількість літер «А» в ньому.  
Вивести ім'я і кількість літер у третьому слові.

### **ВАРІАНТ 2**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу  
" СНУРНИКОВА КАТЕРИНА СЕРГІЇВНА "  
Визначити довжину рядка і кількість літер «К» в ньому.  
Вивести довжини прізвища та імені.

### **ВАРІАНТ 3**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу  
" МАРЧЕНКО ІВАН ГЕННАДІЙОВИЧ "  
Визначити довжину рядка і кількість літер «Н» в ньому.  
Вивести прізвище та ініціали.

### **ВАРІАНТ 4**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу  
" БЄЛОВА ВІКТОРІЯ КОСТЯНТИНІВНА "  
Визначити довжину рядка і кількість літер «В» в ньому.  
Визначити, скільки літер «І» є в імені.

### **ВАРІАНТ 5**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу  
" КРАВЧЕНКО ГАННА ВЛАДИСЛАВІВНА "  
Визначити довжину рядка і кількість літер «А» в ньому.  
Вивести найдовше слово.

### **ВАРІАНТ 6**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу  
" СМИРНОВА МАРГАРИТА ІВАНІВНА "  
Визначити довжину рядка і кількість літер «И» в ньому.  
Вилучити усі літери «А» і «О» з прізвища.

### **ВАРІАНТ 7**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу  
" СТЕПАНОВ МАТВІЙ МИКОЛАЙОВИЧ "  
Визначити довжину рядка і кількість літер «О» в ньому.  
Чи починається хоча б одне слово з літери «М»?

### **ВАРІАНТ 8**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу  
" КЛЮЄВА ВІКТОРІЯ ПЕТРІВНА "

Визначити довжину рядка і кількість літер «А» в ньому.  
Всі літери «І» в імені продублювати.

### **ВАРІАНТ 9**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу  
" СШУРНИКОВА МАРІЯ ОЛЕКСІЇВНА "

Визначити довжину рядка і кількість літер «А» в ньому.  
Вивести ім'я в зворотному порядку.

### **ВАРІАНТ 10**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу  
" ВОЙТЮШЕНКО АНАСТАСІЯ ЮРІЇВНА "

Визначити довжину рядка і кількість літер «Н» в ньому.  
Вивести прізвище в стовпчик.

### **ВАРІАНТ 11**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу  
" ОМЕЛЬЧЕНКО ГАННА ІГОРІВНА "

Визначити довжину рядка і кількість літер «А» в ньому.  
Вивести найкоротше слово.

### **ВАРІАНТ 12**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу  
" НЕСТЕРЕНКО МАРІЯ ІВАНІВНА "

Визначити довжину рядка і кількість літер «А» в ньому.  
Вивести ім'я і кількість літер у прізвищі.

### **ВАРІАНТ 13**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу  
" КАЛИНИЧЕНКО ВАДИМ ЮРІЙОВИЧ "

Визначити довжину рядка і кількість літер «И» в ньому.  
Кожну літеру імені продублювати.

### **ВАРІАНТ 14**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу

**"ПАВЛОВА ОЛЬГА СЕРГІЙВНА"**

Визначити довжину рядка і кількість літер «А» в ньому.  
Вивести прізвище в зворотному порядку.

**ВАРІАНТ 15**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу

**"БАРАНЦЕВИЧ ІВАН РОМАНОВИЧ"**

Визначити довжину рядка і кількість літер «В» в ньому.  
Визначити, скільки літер «А» і «Б» у прізвищі.

**ВАРІАНТ 16**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу

**"РИЖАНКОВА МАРИНА ВАСИЛІВНА"**

Визначити довжину рядка і кількість літер «А» в ньому.  
Вивести третє слово і кількість літер у прізвищі.

**ВАРІАНТ 17**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу

**"ПРОСКУРІНА ВІКТОРІЯ ПАВЛІВНА"**

Визначити довжину рядка і кількість літер «О» в ньому.  
Вивести прізвище і кількість літер по батькові.

**ВАРІАНТ 18**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу

**"КУВАЛДІНА СОФІЯ ПАВЛІВНА"**

Визначити довжину рядка і кількість літер «Н» в ньому.  
У прізвищі замінити всі літери «У» на «О», а всі літери «Д» на «Н».

**ВАРІАНТ 19**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу

**"ЯСЕНЕВ ВОЛОДИМИР АНАТОЛІЙОВИЧ"**

Визначити довжину рядка і кількість літер «А» в ньому.  
Визначити, скільки в тексті голосних літер.

**ВАРІАНТ 20**

Увести прізвище, ім'я і по батькові як одне дане рядкового типу

**"КРОТОВА НАТАЛІЯ ІВАНІВНА"**

Визначити довжину рядка і кількість літер «О» в ньому.  
Визначити, скільки в тексті приголосних літер.

## Тема 8. КЛАСИ

### Об'єктно-орієнтоване програмування

У процедурних мовах, таких як С, програмування орієнтовано на операції.

Всі програми в мові С# є *об'єктно-орієнтованими*.

У мові С елементом програмування є функція (у С# функції називаються методами). У С# елементом програмування є *клас*. Об'єкти, в кінцевому підсумку, створюються з цих класів, і функції інкапсулюються в "рамках" класів як методи.

Програмісти на мові С зосереджуються на написанні функцій. Вони групують операції, які виконують певні завдання, в функцію, після чого функції групуються і утворюють програму. Безсумнівно, в мові С дані важливі, але вони існують в основному для підтримки операцій, які виконуються функціями.

У С# – навпаки: програмісти зосереджені на створенні власних, *визначених користувачем типів*, званих *класами*. Класи ще називаються *типами*, *визначеними програмістом*. Кожен клас містить дані і набір методів для маніпуляцій ними. Компоненти даних класу називаються змінними екземпляра (багато хто використовує термін *поля*). Точно так же, як можна назвати екземпляр вбудованого типу, наприклад `int`, змінної, так і екземпляр типу, визначеного програмістом (тобто клас) називається *об'єктом*. У С# упор зроблений на класи, а не на методи. Програмісти спочатку визначають початковий набір класів, з якого буде розпочато процес проектування. Програмісти використовують ці класи для створення об'єктів, які будуть працювати спільно в рамках цілої системи.

#### **Властивості об'єктно-орієнтованого програмування:**

1. Інкапсуляція (об'єднання в одній структурі даних – класі оголошення даних і методів їх обробки). Об'єкт – це те, що підтримує інкапсуляцію (об'єднує в собі дані і код, який працює з ними).

2. Поліморфізм (можна мати кілька реалізацій одного методу з автоматичним вибором відповідного). Для компілятора поліморфні функції повинні відрізнятися прийнятими параметрами. Ця різниця може бути за їх кількістю або за їх типами.

3. Спадкування – це процес, за допомогою якого один об'єкт може успадковувати основні властивості іншого об'єкта і додавати до них риси, характерні тільки для нього.

При використанні успадкування можна описати об'єкт шляхом визначення того загального класу, до якого він належить, але зі спеціальними рисами, що роблять об'єкт унікальним.

## 8.1. Приклади розв'язання задач

### Задача 8.1. Об'єкт сфера

Написати програму з класом, що описує сферу. Як параметри конструктора передати радіус і щільність речовини сфери. За допомогою методів класу обчислити і з Main() вивести на екран поля, що містять об'єм сфери та її масу:

$$V = \frac{4}{3}\pi r^3; \quad m = V\rho$$

$$\rho = 7870 \text{ кг/м}^3 \text{ – щільність заліза}$$

*Програмний код:*

```
using System;

namespace C311
{
    class Obj
    {
        public double V, M;
        public Obj(double R, double P)
        {
            V = 4.0 / 3.0 * Math.PI * Math.Pow(R, 3.0);
            M = V * P;
        }
    }

    class Program
    {
        static void Result(Obj A)
        {
            Console.WriteLine($" Об'єм сфери = {A.V:F4} м^3\n
                               Маса сфери = {A.M:F3} кг");
            Console.ReadKey(true);
        }
    }
}
```

```

    static void Main(string[] args)
    {
        Obj A = new Obj(0.1, 7870.0);
        Result(A);
    }
}

```

Результати розрахунку подано на рис. 8.1.

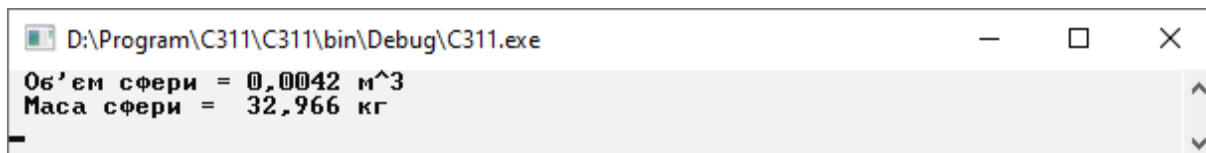


Рисунок 8.1 – Результати розв’язання задачі 8.1

*Пояснення:*

Простір імен *C311* містить два класи *Obj* і *Program*.

### Клас *Obj*

1) `class Obj`

Заголовок класу, *Obj* – ім’я класу.

*Оголошення класу* складається з двох частин: оголошення заголовка класу і оголошення тіла класу.

*Заголовок класу* складається з модифікатора доступу, ключового слова `class` та імені самого класу.

*Тіло класу* – це конструкція, що поміщена у фігурні дужки та містить оголошення полів і методів, які належать класу.

Клас – це тип, що описує облаштування об’єктів.

Клас створюється за допомогою ключового слова `class`. Клас містить змінні, що належать класу (поля), і методи. Члени класу можуть бути закритими в межах класу (специфікатор `private`) або відкритими, тобто доступними за межами класу (`public`). Специфікатор доступу (`public` або `private`) визначає тип дозволеного доступу. Якщо специфікатор доступу відсутній, то оголошуваний член вважається закритим у межах класу. Члени із закритим доступом можуть використовуватися тільки іншими членами їх класу.

Визначення `class` означає створення нового типу даних. У цьому випадку новий тип даних називається *Obj*. За допомогою цього імені можуть бути оголошені об’єкти типу *Obj*. Не слід забувати, що оголошення `class` лише описує тип, але не створює конкретний об’єкт.

Можна сказати, що клас є шаблоном для об'єкта, що описує його структуру і поведження. Поля класу визначають структуру об'єкта, методи класу – його поведження. Клас є типом даних, а об'єкт – змінною цього типу.

*Елементи класу:*

*Константи* класу зберігають незмінні значення, пов'язані з класом.

*Поля* містять дані класу.

*Методи* реалізують обчислення або інші дії, що виконуються класом чи екземпляром.

*Властивості* визначають характеристики класу в сукупності із способами їх задання й отримання, тобто методами запису і читання.

*Конструктори* реалізують дії з ініціалізації екземплярів або класу в цілому.

*Деструкції* визначають дії, які необхідно виконати до того, як об'єкт буде знищений.

*Індексатори* забезпечують можливість доступу до елементів класу за їх порядковим номером.

2) `public double V, M;`

Клас `Obj` містить два дійсні поля даних `V` і `M`, доступних ззовні класу, оскільки вони описані модифікатором `public`.

3)

```
public Obj(double R, double P)
{
    V = 4.0 / 3.0 * Math.PI * Math.Pow(R, 3.0);
    M = V * P;
}
```

Це конструктор, який приймає два параметри `R` і `P` типу `double`. Ці параметри оголошуються в дужках після імені конструктора. Параметрам `R` і `P` передаються значення, які потім використовуватимуться у формулах при обчисленні `V` і `M`.

*Конструктор* – це спеціальний метод об'єкта, що розв'язує задачу початкової ініціалізації полів об'єкта.

Оголошується так:

- для цього методу завжди використовується модифікатор доступу `public`, оскільки конструктор повинен викликатися з коду, визначеного за межами його класу;

- немає типу значення, що повертається (немає навіть `void`);
- ім'я методу збігається з ім'ям класу.

Нижче наведено загальну форму конструктора:  
доступ ім'я\_класу (список параметрів)  
{  
// тіло конструктора  
}

Як правило, конструктор використовується для задання первинних значень змінних екземпляра, визначених у класі.

### Клас Program

Клас Program містить два методи:  
головний метод `static void Main(string[] args)`, з нього розпочинається виконання програми, і метод `static void Result(Obj A)`.

```
1) Obj A = new Obj(0.1, 7870.0);
```

Оголошується об'єкт A типу Obj (створюється об'єкт A як екземпляр класу Obj). Після виконання цього оператора об'єкт A стане екземпляром класу Obj, тобто набуде «фізичної реальності».

*У загальному вигляді оголошення набувають форми:*

```
ClassName objectReference = new ClassName (arguments);
```

де `objectReference` – посилання відповідного типу даних;

`new` указує на те, що створюється об'єкт;

`ClassName` вказує тип нового об'єкта (та ім'я конструктора, що викликається);

`arguments` – задає розділений комами список значень, використовуваних конструктором для ініціалізації об'єкта.

Цей рядок оголошення виконує три функції. По-перше, оголошується змінна A, що належить до типу класу Obj. По-друге, створюється конкретна, фізична копія об'єкта. Це робиться за допомогою оператора `new`. І нарешті, змінній A привласнюється посилання на цей об'єкт. Таким чином, після виконання цього рядка оголошена змінна A посилається на об'єкт типу Obj.

Оператор `new` динамічно (тобто під час виконання) розподіляє пам'ять для об'єкта і повертає посилання для нього, яке потім зберігається в змінній.

При створенні екземпляра А класу `Obj` використовується конструктор, що ініціалізує значення полів значеннями аргументів.

Конструктор `Obj()` викликається в операторі `new` для об'єкта А, привласнюючи змінній його екземпляра результати обчислення за формулами `A.V` і `A.M`.

Кожен об'єкт типу `Obj` міститиме свої копії змінних екземпляра `V` і `M`. Для доступу до цих змінних служить оператор доступу до члена класу, який прийнято називати оператором-точкою. Оператор-точка зв'язує ім'я об'єкта з ім'ям члена класу.

Нижче наведено загальну форму оператора-точки:

Об'єкт.член

2) `Result(A)`;

Викликається статичний метод `Result()`, і йому передається як параметр А об'єкт типу `Obj`. Статичний метод `Result()` за допомогою методу `WriteLine()` виводить результати розрахунку `A.V` і `A.M` (імена полів з'єднуються з іменами об'єктів, що утримують їх, точкою).

## Задача 8.2

Тіло кинути під кутом  $\alpha$  до горизонту зі швидкістю  $v_0$ . Знайти час руху  $t$ , дальність польоту  $L$ , максимальну висоту підйому  $h$ , швидкість тіла у момент падіння  $v_k$ .

Клас, що описує об'єкти, які кидають, інкапсулює дані об'єкта та методи, які їх обробляють. Конструктор класу повинен ініціалізувати вхідні дані і викликати методи, що обчислюють шукані дані.

$$t = \frac{2v_0 \sin \alpha}{g} \quad \text{Час руху}$$

$$t_1 = \frac{v_0 \sin \alpha}{g} \quad \text{Час підйому}$$

$$h = \frac{v_0^2 \sin^2 \alpha}{2g} \quad \text{Висота підйому}$$

$$L = \frac{v_0^2 \sin 2\alpha}{g} \quad \text{Дальність польоту}$$

$$v_k = v_0 \quad \text{Швидкість тіла у момент падіння}$$

*Програмный код:*

```
using System;

namespace C312
{
    class Obj
    {
        public double v0, vk, L, t, t1, h, alfa;
        public const double g = 9.8;
        public Obj(double a, double v)
        {
            alfa = a * Math.PI / 180; v0 = v;
            vk = v0; t1 = Time1(); t = Time();
            L = Length(); h = Height();
        }

        public double Length()
        {
            return v0 * v0 * Math.Sin(2 * alfa) / g;
        }

        public double Height()
        {
            return v0 * v0 * Math.Pow(Math.Sin(alfa), 2.0) / 2.0 / g;
        }

        public double Time1()
        {
            return v0 * Math.Sin(alfa) / g;
        }

        public double Time()
        {
            return Time1() * 2;
        }
    }

    class Program
    {
        static void Result(int n, Obj R)
        {
```

```

        Console.WriteLine($"A{n} пролетів {R.L:F2} м
            за {R.t:F2} с, піднявшись на {R.h:F2} м
            (v0= {R.v0:F2}, a= {R.alfa:F2})");
    }

    static void Main(string[] args)
    {
        Obj A1 = new Obj(45.0, 15.0);
        Obj A2 = new Obj(15.0, 45.0);
        Result(1, A1);
        Result(2, A2);
        Console.ReadLine();
    }
}

```

Результати розрахунку подано на рис. 8.2.

Рисунок 8.2 – Результати розв’язання задачі 8.2

### Задача 8.3

Написати програму з класом, який дозволяє:

- 1) створити одновимірний масив;
- 2) обчислювати суму елементів одновимірного масиву  $A(n)$ ;
- 3) обчислювати суму елементів одновимірного масиву  $A(n)$ , починаючи із заданого елемента;
- 4) обчислювати середнє арифметичне елементів одновимірного масиву  $A(n)$ ;
- 5) обчислювати добуток елементів одновимірного масиву  $A(n)$ ;
- 6) обчислювати кількість додатних елементів одновимірного масиву  $A(n)$ ;
- 7) обчислювати суму додатних елементів одновимірного масиву  $A(n)$ ;
- 8) обчислювати добуток додатних елементів одновимірного масиву  $A(n)$ ;
- 9) виводити елементи масиву на екран.

*Програмний код:*

```
using System;

namespace C313
{
    class Obj
    {
        int n; // дані класу
        double[] A;
        public Obj(int k1) // конструктор
        {
            n = k1;
            A = new double[k1];
            for (int i = 0; i < n; i++)
            {
                Console.Write($" Введіть A[{i}]= ? ");
                A[i] = Convert.ToInt32(Console.ReadLine());
            }
        }

        ~Obj() //деструктор
        {
            Console.WriteLine("Деструктор");
        }

        // метод класу
        public void SUMM(out double s)
        {
            // обчислює суму елементів масиву
            s = 0;
            for (int i = 0; i < n; i++)
                s = s + A[i];
        }

        // метод класу
        public void SUM2(int q, ref double szad)
        {
            // обчислює суму елементів,
            // починаючи із заданого елемента
            double s = 0;
        }
    }
}
```

```

    if (q >= n)
        q = -5;
    else
    {
        for (int i = q; i < n; i++)
            s = s + A[i];
        szad = s;
    }
}

// метод класу
public void SRED(out double sr)
{
    // обчислює середнє арифметичне елементів масиву
    double s = 0;
    for (int i = 0; i < n; i++)
        s = s + A[i];
    sr = s / n;
}

// метод класу
public void PROIZ(out double p)
{
    // обчислює добуток елементів масиву
    p = 1;
    for (int i = 0; i < n; i++)
        p = p * A[i];
}

// метод класу
public void KOL(out int K)
{
    // обчислює кількість додатних елементів масиву
    K = 0;
    for (int i = 0; i < n; i++)
    {
        if (A[i] > 0)
            K = K + 1;
    }
}

```

```

// метод класу
public void SUMPOL(out double ss)
{
    // обчислює суму додатних елементів масиву
    double s = 0;
    for (int i = 0; i < n; i++)
    {
        if (A[i] > 0)
            s = s + A[i];
    }
    ss = s;
}

// метод класу
public void PROIZPOL(out double pp)
{
    // обчислює добуток додатних елементів масиву
    double p = 1;
    for (int i = 0; i < n; i++)
    {
        if (A[i] > 0)
            p = p * A[i];
    }
    pp = p;
}

// метод класу
public int get_n()
{
    // повертає n-кількість елементів масиву
    return n;
}

// метод класу
public void OUT()
{
    // виводить елементи масиву на екран
    for (int i = 0; i < n; i++)
        Console.WriteLine($" Елемент A[{i}] = {A[i]}");
}
}

```

```

class Program
{
    static void Main(string[] args)
    {
        int L, m, K1;
        double s1, szad2, sr1, p1, ss1, pp1;
        Console.Write(" Введіть кількість елементів? ");
        m = Convert.ToInt32(Console.ReadLine());
        Obj B = new Obj(m); // створення екземпляра класу
        L = B.get_n(); // виклик методу класу
        Console.WriteLine("\n Елементів: " + L);
        Console.WriteLine();
        B.OUT();
        B.SUMM(out s1);
        Console.WriteLine("\n Сума усіх елементів масиву = " +
s1);

        szad2 = 1; // привласнення значення szad2 обов'язково
        B.SUM2(3, ref szad2);
        Console.WriteLine("\n Сума елементів масиву, починаючи з
3 - го елемента = " + szad2);
        B.SRED(out sr1);
        Console.WriteLine("\n Середнє арифметичне елементів
масиву = " + sr1);
        B.PROIZ(out p1);
        Console.WriteLine("\n Добуток всіх елементів масиву = "
+ p1);
        B.KOL(out K1);
        Console.WriteLine("\n Кількість додатних елементів
масиву = " + K1);
        B.SUMPOL(out ss1);
        Console.WriteLine("\n Сума додатних елементів масиву = "
+ ss1);
        B.PROIZPOL(out pp1);
        Console.WriteLine("\n Добуток додатних елементів масиву
= " + pp1);
        Console.ReadLine();
    }
}

```

Результати розрахунку подано на рис. 8.3.

```
D:\Program\C313\C313\bin\Debug\C313.exe
Введіть кількість елементів? 10
Введіть A[0]= ? 1
Введіть A[1]= ? -2
Введіть A[2]= ? 3
Введіть A[3]= ? 4
Введіть A[4]= ? -5
Введіть A[5]= ? 6
Введіть A[6]= ? -7
Введіть A[7]= ? 8
Введіть A[8]= ? -9
Введіть A[9]= ? 10

Елементів: 10

Елемент A[0]= 1
Елемент A[1]= -2
Елемент A[2]= 3
Елемент A[3]= 4
Елемент A[4]= -5
Елемент A[5]= 6
Елемент A[6]= -7
Елемент A[7]= 8
Елемент A[8]= -9
Елемент A[9]= 10

Сума усіх елементів масиву = 9
Сума елементів масиву, починаючи з 3 - го елемента = 7
Середнє арифметичне елементів масиву = 0,9
Добуток всіх елементів масиву = 3628800
Кількість додатних елементів масиву = 6
Сума додатних елементів масиву = 32
Добуток додатних елементів масиву = 5760
```

Рисунок 8.3 – Результати розв’язання задачі 8.3

*Пояснення:*

Наш клас `Obj` має як дані змінну `n` і масив `A`, обоє вони мають атрибут доступу `private`.

Функція, ім’я якої збігається з ім’ям класу, є конструктором. Природно, що конструктор повинен мати атрибут доступу `public`, інакше неможливо створити екземпляри класу (об’єкти). Конструктор може мати формальні параметри за загальними правилами, але він не може мати зворотного значення, і вказувати тип зворотного значення заборонено. Допускається наявність в одному класі більше одного конструктора, але у них повинен бути різний склад формальних параметрів. Вибір конструктора в такому випадку здійснюється на основі фактичних параметрів. У конструкторі зазвичай пишуть операції ініціалізації і введення початкових даних. Конструктори, як і інші функції у складі класу, можуть працювати з усіма змінними і масивами свого класу, які є як би глобальними змінними для них.

Функція, ім'я якої має структуру `~ ім'я_класу()`, називається *деструктором*. Деструктори визначають дії, які необхідно виконати до того, як об'єкт буде знищений. У C# самому викликати деструктор неможливо. Деструкція викликається автоматично програмою, що називається «збирач сміття». У C# немає операторів звільнення пам'яті, це виконує та сама програма «збирач сміття», яка видаляє об'єкти, що стали непотрібними. Деструкція повинна забезпечити коректну ліквідацію об'єкта.

Далі слідує інші функції – члени класу.

У C# дозволено перевантаження методу: два або більше методів у одному класі (у тому числі і конструктор) можуть мати одне і те саме ім'я, за умови, що у них різний склад (їх кількість і/або типи) формальних параметрів. Такі методи називають *перевантаженими*. Відмінність тільки в типі зворотного значення недостатня для перевантаження, і такі методи не вважаються перевантаженими.

У класі `Program` розглянемо головну функцію `Main()`.

У ній спочатку вводиться кількість елементів `m`:

```
Console.Write("Введіть кількість елементів:");  
m=Convert.ToInt32(Console.ReadLine());
```

Потім створюється екземпляр класу

```
Obj B = new Obj(m);
```

Це означає запуск конструктора класу і в нашому випадку виділення пам'яті під масив і уведення цього масиву.

Далі слідує виклики методів класу традиційним способом. Зверніть увагу на використання модифікаторів **out**, **ref**.

#### Задача 8.4

Написати програму з класом для роботи з двовимірним масивом дійсних чисел  $A(n, m)$ .

*Клас забезпечить:*

1. Створення двовимірного масиву.
2. Виведення елементів двовимірного масиву в консольне вікно.
3. Знаходження суми елементів кожного рядка матриці.
4. Знаходження добутку елементів кожного рядка матриці.
5. Знаходження суми елементів кожного стовпця матриці.
6. Знаходження добутку елементів кожного стовпця матриці.

*Програмний код:*

```
using System;

namespace C314
{
    class Obj
    {
        int n, m; // дані класу
        double[,] A;
        public Obj(int k1, int k2) // конструктор
        {
            int i, j;
            n = k1;
            m = k2;
            A = new double[k1, k2];
            Random rnd = new Random();
            // Заповнення матриці A
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < m; j++)
                {
                    A[i, j] = rnd.NextDouble() * 10;
                }
            }
        }

        //метод класу
        public void OUT()
        {
            // виводить елементи масиву на екран
            int i, j;
            Console.WriteLine(" Масив A: ");
            for (i = 0; i < n; i++)
            {
                for (j = 0; j < m; j++)
                {
                    Console.Write($" {A[i,j],9:F2}");
                }
                Console.WriteLine();
            }
        }
    }
}
```

```

//метод класу
public void SumSTROKA()
{
    int i, j;
    // Обчислення суми елементів
    // кожного рядка матриці
    double s;
    for (i = 0; i < n; i++)
    {
        s = 0;
        for (j = 0; j < m; j++)
        {
            s = s + A[i, j];
        }
        Console.WriteLine($" Сума елементів {i} рядка =
{s:F3}");
    }
}

//метод класу
public void ProizSTROKA()
{
    int i, j;
    // Знаходження добутку елементів
    // кожного рядка матриці
    double p;
    for (i = 0; i < n; i++)
    {
        p = 1;
        for (j = 0; j < m; j++)
        {
            p = p * A[i, j];
        }
        Console.WriteLine($" Добуток елементів {i} рядка =
{p:F3}");
    }
}

//метод класу
public void SumSTOLBEZ()
{
    int i, j;
    // Знаходження суми елементів

```

```

        // кожного стовпця матриці
        double s;
        for (j = 0; j < m; j++)
        {
            s = 0;
            for (i = 0; i < n; i++)
            {
                s = s + A[i, j];
            }
            Console.WriteLine($" Сума елементів {j} стовпця =
{s:F3}");
        }
    }

    //метод класу
    public void ProizSTOLBEZ()
    {
        int i, j;
        // Знаходження добутку елементів
        // кожного стовпця матриці
        double p;
        for (j = 0; j < m; j++)
        {
            p = 1;
            for (i = 0; i < n; i++)
            {
                p = p * A[i, j];
            }
            Console.WriteLine($" Добуток елементів {j} стовпця =
{p:F3}");
        }
    }
}

class Program
{
    static void Main(string[] args)
    {
        int q1; // кількість рядків матриці
        int q2; // кількість стовпців матриці
        Console.Write(" Введіть кількість рядків ? ");
        q1 = Convert.ToInt32(Console.ReadLine());
        Console.Write(" Введіть кількість стовпців ? ");
        q2 = Convert.ToInt32(Console.ReadLine());
    }
}

```

```

Obj B = new Obj(q1, q2); // створення екземпляра класу

// Виведення елементів матриці в консольне вікно
B.OUT();
Console.WriteLine();

// Знаходження суми елементів кожного рядка
B.SumSTROKA();
Console.WriteLine();

// Знаходження добутку елементів кожного рядка
B.ProizSTROKA();
Console.WriteLine();

// Знаходження суми елементів кожного стовпця
B.SumSTOLBEZ();
Console.WriteLine();

// Знаходження добутку елементів кожного стовпця
B.ProizSTOLBEZ();
Console.ReadLine();
    }
}
}

```

Результати розрахунку наведено на рис. 8.4.

```

D:\Program\C314\C314\bin\Debug\C314.exe
Введіть кількість рядків ? 4
Введіть кількість стовпців ? 5
Масив A:
  4,41    7,11    4,98    8,32    3,69
  0,26    8,32    2,19    6,56    6,78
  6,65    5,89    2,29    7,42    9,55
  9,65    6,67    7,30    0,46    4,19

Сума елементів 0 рядка = 28,510
Сума елементів 1 рядка = 24,105
Сума елементів 2 рядка = 31,802
Сума елементів 3 рядка = 28,273

Добуток елементів 0 рядка = 4795,151
Добуток елементів 1 рядка = 208,011
Добуток елементів 2 рядка = 6352,849
Добуток елементів 3 рядка = 906,180

Сума елементів 0 стовпця = 20,970
Сума елементів 1 стовпця = 27,987
Сума елементів 2 стовпця = 16,758
Сума елементів 3 стовпця = 22,760
Сума елементів 4 стовпця = 24,215

Добуток елементів 0 стовпця = 72,664
Добуток елементів 1 стовпця = 2322,898
Добуток елементів 2 стовпця = 182,129
Добуток елементів 3 стовпця = 186,363
Добуток елементів 4 стовпця = 1002,266

```

Рисунок 8.4 – Результати розв’язання задачі 8.4

## Створення класів-спадкоємців

*Синтаксис успадкування* такий: при описі класу-нащадка його клас-предок вказується через двокрапку.

Приклад визначення класу-предка `Dad` і класу-нащадка `Son`:

```
public class Dad{ }  
public class Son: Dad{ }
```

При ініціалізації полів об'єктів класу-спадкоємця необхідно також ініціалізувати і поля базового класу.

Ініціалізація полів здійснюється з використанням конструктора.

Передача керування конструктору базового класу при створенні об'єкта-представника похідного класу здійснюється за допомогою конструкції:

```
...(...):base(...){...},
```

яка розташовується в оголошенні конструктора класу-спадкоємця між заголовком конструктора і тілом. Після ключового слова `base` у дужках розташовується список значень параметрів конструктора базового класу. Очевидно, що вибір відповідного конструктора визначається типом значень у списку (можливо, порожньому) параметрів.

*Приклад:*

```
public class Dad  
{  
    int a;  
    public Dad(int s);  
}  
public class Son: Dad  
{  
    public Son(int k):base(k) {}  
}
```

Якщо у базового класу не оголошено жодного конструктора (залишений конструктор за замовчуванням) або оголошений конструктор не має параметрів, тоді конструкцію `base` можна не використовувати: за її відсутності керування передається конструктору без параметрів.

Проте при виклику конструктора можна передавати керування не лише конструктору базового класу, але і іншому конструктору цього класу. Це зручно в тих випадках, коли необхідно створити безліч об'єктів, що розрізняються між собою якимось чином, але мають якусь загальну частину. Тоді для реалізації загальної частини можна написати якийсь загальний конструктор, а вже в інших конструкторах, що виконують більш

детальне налаштування об'єкта, викликати загальний конструктор. Передача керування власному конструктору аналогічна описаній вище, тільки замість ключового слова `base` використовується ключове слово `this`.

*Приклад:*

```
public class Dad
{
    int a;
    public Dad(int s);
}
public class Son: Dad
{
    public Son(int k):base(k) {}
    public Son():this(10) {}
}
```

### **Задача 8.5**

Описати базовий клас `arr`, який дозволяє створити одновимірний масив і вивести елементи масиву в консольне вікно.

Описати клас `proc1` – це клас-спадкоємець базового класу `arr`, він призначений для уведення числа-межі й обчислення суми тих елементів масиву, значення елементів яких перевищують число-межу.

Написати програму, яка використовує ці класи.

**Спосіб 1** – конструктори базового класу і класу-предка без параметрів

*Програмний код:*

```
using System;

namespace C315
{
    class arr
    {
        protected int[] k; // атрибут доступу protected потрібний
        // для забезпечення доступу з класів - спадкоємців
        public arr()
        { // конструктор
            int n;
            Console.Write(" Кількість елементів ? ");
            n = Convert.ToInt32(Console.ReadLine());
            k = new int[n];
        }
    }
}
```

```

        for (int i = 0; i < n; i++)
        {
            Console.WriteLine($" K[{i}]= ? ");
            k[i] = Convert.ToInt32(Console.ReadLine());
        }
    }

    public void output()
    {
        Console.WriteLine();
        Console.WriteLine(" Елементи масиву:");
        for (int i = 0; i < k.Length; i++)
            Console.WriteLine($" K[{i}]= {k[i]}");
    }
}

class proc1 : arr // задаємо базовий клас arr
{
    int q;
    public proc1()
    { // конструктор класу-спадкоємця
        Console.WriteLine(" Введіть число-межу ? ");
        q = Convert.ToInt32(Console.ReadLine());
    }

    public int sum()
    {
        int s = 0;
        for (int i = 0; i < k.Length; i++)
            if (k[i] > q) s = s + k[i];
        return s;
    }
}

class Program
{
    static void Main(string[] args)
    {
        proc1 My = new proc1(); //1
        int s1;
        My.output();           // звернення до методу предка
        s1 = My.sum();         // звернення до власного методу,
        // аналогічно можна звертатися і до властивостей предка
    }
}

```

```

        Console.WriteLine($"Сума елементів масиву, які більші
                           від уведеної межі = {s1}");
        Console.ReadLine();
    }
}
}

```

Результати розрахунку подано на рис. 8.5.

```

D:\Program\C315\C315\bin\Debug\C315.exe
Кількість елементів ? 10
K[0]= ? 1
K[1]= ? 2
K[2]= ? 5
K[3]= ? 7
K[4]= ? 8
K[5]= ? 12
K[6]= ? 15
K[7]= ? 17
K[8]= ? 19
K[9]= ? 21
Введіть число-межу ? 8

Елементи масиву:
K[0]= 1
K[1]= 2
K[2]= 5
K[3]= 7
K[4]= 8
K[5]= 12
K[6]= 15
K[7]= 17
K[8]= 19
K[9]= 21
Сума елементів масиву, які більші від уведеної меж? = 84

```

Рисунок 8.5 – Результати розв’язання задачі 8.5 (Спосіб 1)

*Пояснення:*

У С# допускається просте спадкоємство: кожен клас може мати тільки одного предка. Використовуючи спадкоємство, можна створити базовий клас, який визначає характеристики, властиві безлічі пов’язаних об’єктів. Цей клас потім може бути успадкований іншими класами з додаванням у кожен з них своїх особливостей. Рівнозначні терміни: базовий клас – клас-спадкоємець; батьківський клас – дочірній клас; клас-предок – клас-спадкоємець.

Створимо як приклад базовий клас для обробки масиву `arr`, що включає визначення масиву, його введення і виведення. До елементів базового класу з атрибутом доступу `private` немає доступу з класів-спадкоємців, вони, таким чином, не наслідуються. Тому рекомендують дати елементам базового класу атрибут доступу `protected`.

Конструктор базового класу призначений для уведення кількості елементів масиву і уведення самих елементів. У ньому також проводиться виведення елементів масиву в консольне вікно за допомогою методу `output()`.

Клас `proc1` – це клас-спадкоємець базового класу `arr`, його конструктор призначений для уведення числа-межі, а метод `sum()` обчислює суму тих елементів масиву, які більші числа-межі. Клас-спадкоємець включає усі дані свого предка (за винятком даних з атрибутом доступу `private`).

У класі `Program`

1. Створюється екземпляр класу `proc1`

```
proc1 My = new proc1();
```

При створенні екземпляра класу, що має предка, запускаються усі конструктори: в першу чергу, конструктор базового класу і потім конструктор класу-спадкоємця. У нашому випадку це означає, що буде здійснено уведення спочатку масиву і слідом за ним – межі. За наявності більшої кількості рівнів наслідування підряд будуть запусчені конструктори всіх рівнів ієрархії, починаючи з базового.

У цьому випадку конструктори не мають формальних параметрів, кожен конструктор незалежно від інших виконує свої оператори.

2. Звертаємося до методу предка для друку результатів

```
My.output();
```

3. Звертаємося до власного методу для знаходження суми елементів масиву, які більші числа-межі

```
s1 = My.sum();
```

4. Виводимо значення суми в консольне вікно

```
Console.WriteLine($"Сума елементів масиву, які більші від уведеної межі = {s1}");
```

**Спосіб 2** – конструктори базового класу і класу-предка з параметрами

*Програмний код:*

```
using System;
```

```
namespace C316
```

```
{
```

```
    class arr
```

```
    {
```

```
        protected int[] k; // атрибут доступу protected
```

```

// потрібний для забезпечення доступу з класів-спадкоємців
public arr(int p)
{ // конструктор
  k = new int[p];
  for (int i = 0; i < p; i++)
  {
    Console.Write($"K[{i}]= ? ");
    k[i] = Convert.ToInt32(Console.ReadLine());
  }
}

public void output()
{
  Console.WriteLine();
  Console.WriteLine("Елементи масиву ");
  for (int i = 0; i < k.Length; i++)
    Console.WriteLine($"K[{i}]= {k[i]}");
}
}

class proc1 : arr // задаємо базовий клас arr
{
  int q;
  public proc1(int k1, int k2) : base(k1)
  { // конструктор класу-спадкоємця
    q = k2;
  }

  public int sum()
  {
    int s = 0;
    for (int i = 0; i < k.Length; i++)
      if (k[i] > q) s = s + k[i];
    return s;
  }
}

class Program
{
  static void Main(string[] args)
  {
    int s1;
    // звернення до конструктора з параметрами
  }
}

```

```

        proc1 Myaa = new proc1(10, 8);
        Myaa.output();          // звернення до методу предка
        s1 = Myaa.sum();        // звернення до власного методу,
        // аналогічно можна звертатися і до властивостей предка
        Console.WriteLine($"Сума елементів масиву, які більші
                               від заданої межі = {s1}");

        Console.ReadLine();
    }
}
}

```

Результати розрахунку подано на рис. 8.6.

```

D:\Program\C316\C316\bin\Debug\C316.exe
K[0]= ? 1
K[1]= ? 2
K[2]= ? 5
K[3]= ? 7
K[4]= ? 8
K[5]= ? 12
K[6]= ? 15
K[7]= ? 17
K[8]= ? 19
K[9]= ? 21

Елементи масиву
K[0]= 1
K[1]= 2
K[2]= 5
K[3]= 7
K[4]= 8
K[5]= 12
K[6]= 15
K[7]= 17
K[8]= 19
K[9]= 21
Сума елементів масиву, які більші від заданої межі = 84

```

Рисунок 8.6 – Результати розв’язання задачі 8.5 (Спосіб 2)

*Пояснення:*

1. Забезпечимо передачу параметра конструктору класу-предка за допомогою списку ініціалізації в конструкторі класу-спадкоємця

```

public proc1(int k1, int k2): base(k1)
{
    q = k2;
}

```

Запис `base(k1)` означає, що конструктору базового класу як фактичний параметр буде передано значення `k1`.

2. У класі `Program` в головному методі звертаємося до конструктора з параметрами

```

proc1 Myaa = new proc1(10, 8);

```

де 10 – це кількість елементів масиву;  
8 – число-межа

3. Звертаємося до методу предка для друку результатів

```
Myaa.output();
```

4. Звертаємося до власного методу для знаходження суми елементів масиву, які більші від числа-межі

```
s1 = Myaa.sum();
```

5. Виводимо значення суми в консольне вікно

```
Console.WriteLine($"Сума елементів масиву, які більші від заданої межі = {s1}");
```

## 8.2 Завдання для виконання лабораторних робіт

### Задача 1. Об'єкт труба

Написати програму з класом, що описує циліндричну трубу. Увести з клавіатури і як параметри конструктора передати радіус, товщину стінки, довжину труби. Матеріалом труби прийняти залізо, щільність задати константою  $\rho = 7870 \text{ кг/м}^3$ . За допомогою методів класу обчислити і з `Main()` вивести на екран поля, що містять об'єм, займаний матеріалом труби, і її масу.

$V1 = \pi R^2 L$  – об'єм циліндра зовнішній,

$V2 = \pi (R - T)^2 L$  – об'єм циліндра внутрішній,

$V3 = \pi R^2 L - \pi (R - T)^2 L$  – об'єм, займаний матеріалом труби,

$m = V3 \cdot \rho$  – маса труби.

### Задача 2

Написати програму з класом для знаходження максимального елемента одновимірного масиву і його індексу.

### Задача 3

Написати програму з класом для роботи з двовимірним масивом дійсних чисел  $A(n, n)$  – матриця квадратна.

*Клас має забезпечити:*

1. Створення двовимірного масиву.
2. Виведення елементів двовимірного масиву в консольне вікно.
3. Обчислення суми елементів, розташованих на головній діагоналі матриці.
4. Обчислення суми елементів, розташованих над головною діагоналлю матриці.
5. Обчислення суми елементів, розташованих під головною діагоналлю матриці.
6. Створення вектора з елементів головної діагоналі.

## Тема 9. СТРУКТУРИ, ПЕРЕРАХУВАННЯ

### 9.1 Приклади розв'язання задач

**Задача 9.1.** Описати структуру з ім'ям `Stud`, яка містить такі поля: рядкові поля імені, групи, адреси і номер телефону, поля дати народження і дати вступу, а також поточного середнього бала успішності. Вивести у консольне вікно дані о студентах, які мають середній бал вище 4.5. Указати вік студентів.

*Програмний код:*

```
using System;

namespace C114
{
    struct Stud
    {
        public string Name, Group, Addr, PhoneNum;
        public DateTime BirthDate, EntryDate;
        public double SrBal;
        public Stud(string n, string g, string a, string p,
            DateTime b, DateTime e, double c)
        {
            Name = n; Group = g; Addr = a;
            PhoneNum = p; BirthDate = b;
            EntryDate = e; SrBal = c;
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Stud[] X = new Stud[30];
            X[0].Name = "Максименко Петро Іванович";
            X[0].Group = "ХТ-119к";
            X[0].Addr = "вул. Киргизська 3,8";
            X[0].PhoneNum = "0504323897";
            X[0].BirthDate = new DateTime(2001, 5, 28);
            X[0].EntryDate = new DateTime(2019, 7, 15);
            X[0].SrBal = 4.3;
            X[1] = new Stud("Ніколаєнко Олег Михайлович",
                "ХТ-119к", "вул. Іллінська 5,2", "0503897512",
                new DateTime(2000, 10, 17),
```



середній бал успішності. Далі наводиться ініціалізація першого елемента за допомогою конструктора.

```
X[1] = new Stud("Ніколаєнко Олег Михайлович",
    "ХТ-119к", "вул. Іллінська 5,2", "0503897512",
    new DateTime(2000, 10, 17),
    new DateTime(2019, 7, 15), 4.8);
```

Потрібно зауважити, що необхідно указати значення для всіх полів.

```
new DateTime(2000, 10, 17)
```

Задається дата народження (рік, місяць, день).

Нижче у циклі розраховується вік як різниця між роком народження і поточною датою. Якщо віднімання отриманого року з поточної дати дає дату меншу, ніж дата народження, то значення віку зменшується на одиницю. Ім'я та вік виводяться за допомогою рядка формату.

Для отримання поточної дати використовується властивість `Today` в `DateTime`

```
DateTime TD = DateTime.Today;
```

Приклад виведення TD в консольне вікно

```
15.01.2021 0:00:00
```

Властивість `Year` в `DateTime` використовується для отримання року.

Нижче у таблиці 9.1 наведено дані для заповнення масиву структур `Stud`.

Таблиця 9.1 – Дані для заповнення масиву структур `Stud`

Ім'я (Name)	Група (Group)	Адреса (Addr)	Номер телефону (PhoneNum)	Дата народження (рік, місяць, день) (BirthDate)	Дата вступу (рік, місяць, день) (EntryDate)	Поточний середній бал успішності (SrBal)
Максименко Петро Іванович	ХТ-119к	вул. Киргизська 3,8	0504323897	2001, 5, 28	2019, 7, 15	4.3
Ніколаєнко Олег Михайлович	ХТ-119к	вул. Іллінська 5,2	0503897512	2000, 10, 17	2019, 7, 15	4.8
Сокоренко Андрій Вадимович	ХТ-119к	вул. Волонтерська 8,3	0673672935	2001, 6, 12	2019, 7, 15	4.6
Кравченко Марія Павлівна	ХТ-119к	вул. Героїв праці 15,6	0667111564	2001, 1, 5	2019, 7, 15	4.4
Нестеренко Ольга Валентинівна	ХТ-119к	вул. Полтавський Шлях 150,21	0963026380	2000, 11, 3	2019, 7, 15	4.7

**Задача 9.2.** Програма демонструє застосування структури для зберігання інформації про книгу.

*Програмний код:*

```
using System;

namespace C115
{
    // Визначити структуру
    struct Book
    {
        public string Author;
        public string Title;
        public int Copyright;
        public int stran;

        public Book(string a, string t, int c, int s)
        {
            Author = a;
            Title = t;
            Copyright = c;
            stran = s;
        }
    }

    // Застосування структури
    class Program
    {
        static void Main(string[] args)
        {
            // Створюється масив структур
            Book[] X = new Book[30];
            X[0].Author = "Павловская Т.А.";
            X[0].Title = "С#. Программирование на языке высокого уровня";
            X[0].Copyright = 2013;
            X[0].stran=430;
            X[1] = new Book("Герберт Шилдт", "Полный справочник по
                               С# 4.0", 2010, 900);
            X[2]= new Book("Лукин С.Н.", "Понятно о Visual Basic.Net.
                               Самоучитель. В трёх томах", 2005, 800);
            X[3] = new Book("Гусева О.Л.", "Практикум по Visual
                               Basic", 2007, 544);

            for (int i = 0; i <= 3; i++)
```

```

        Console.WriteLine(X[i].Author + " " + X[i].Title + " "
            + X[i].Copyright + "г " + X[i].stran + "с");
    Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 9.2.

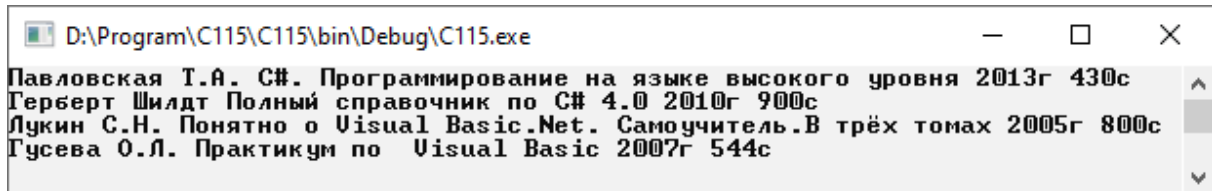


Рисунок 9.2 – Результати розв’язання задачі 9.2

### Задача 9.3. Приклад застосування переліку Colors.

*Програмний код:*

```

using System;

namespace C113
{
    class Program
    {
        enum Colors
        {
            Black, Blue, Green, Cyan, Red, Magenta, Brown, White,
            Gray, LtBlue, LtGreen, LtCyan, LtRed, LtMagenta,
            Yellow, LtWhite
        }

        static void Main(string[] args)
        {
            string[] NColors = {"Чорний", "Синій", "Зелений", "Ціан",
                "Червоний", "Малиновий", "Коричневий", "Білий", "Сірий",
                "Св.Синій", "Св.Зелений", "Св.Ціан", "Св.Червоний",
                "Св.Малиновий", "Жовтий", "Яск.Білий"};
            for (Colors i = Colors.Black; i <= Colors.LtWhite; ++i)
                Console.WriteLine($"{(int)i,2} {i,10}
                    {NColors[(int)i],12}");
            Console.ReadLine();
        }
    }
}

```

Результати розрахунку подано на рис. 9.3.

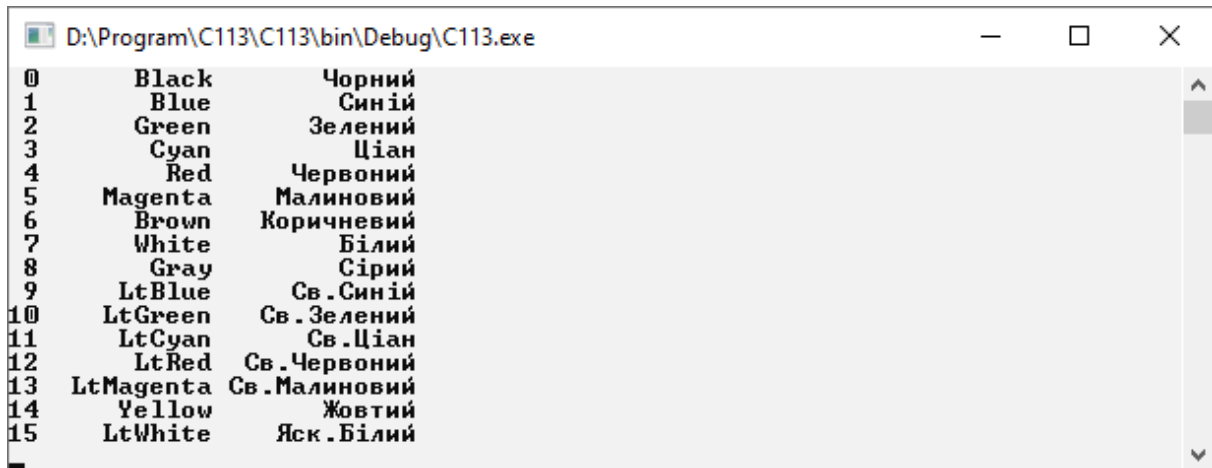


Рисунок 9.3 – Результати розв’язання задачі 9.3

*Пояснення:* У задачі 9.3 наведено приклад переліку `Colors`. У програмі заданий рядковий масив з описом кольорів. У циклі, де лічильником циклу є змінна переліченого типу `Colors`, виводиться на консоль список порядкового номера кольору, ім’я поточного елемента списку переліку й елемент рядкового масиву як опису поточного кольору. У рядку формату, крім номера параметра, через кому вказується розмір поля, що формує таблицю. Вирівнювання в полі проводиться по правому краю, тому що розмір поля додатний.

За замовчуванням номер кожного наступного елемента списку переліку більше попереднього на одиницю. Нумерація починається з нуля. У середині списку будь-якому елементу можна надати ціле число – номер більший, ніж номер за замовчуванням. Нумерацію можна починати не з нуля, а наприклад із двох.

*Приклад:*

```
enum Colors
{
    Black = 2, Blue, Green, Cyan, Red, Magenta, Brown, White,
    Gray, LtBlue, LtGreen, LtCyan, LtRed, LtMagenta, Yellow, LtWhite
}
```

У цьому прикладі всі наступні після `Black` елементи будуть мати номер на одиницю більше від попереднього. Задавати номер елемента можна також і усередині списку.

*Приклад:*

```
enum Colors
{
    Black, Blue, Green, Cyan, Red, Magenta, Brown, White, Gray = 31,
    LtBlue, LtGreen, LtCyan, LtRed, LtMagenta, Yellow, LtWhite
}
```

Якщо номер елемента `White` дорівнює семи, то номер елемента `Gray` рівний 31 і, відповідно, усі наступні за ним елементи мають номер, на одиницю більше від попереднього.

## Задача 9.4. Приклад застосування переліку Apple

*Програмний код:*

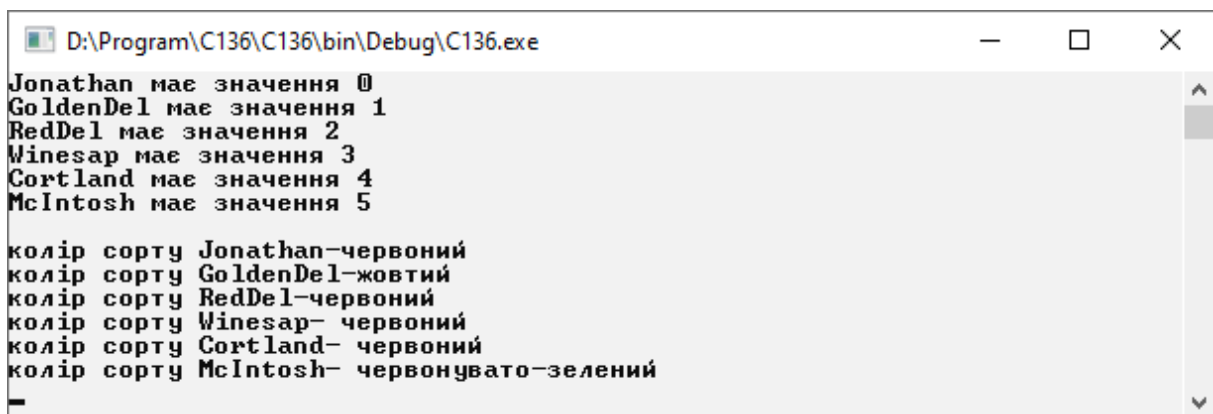
```
using System;

namespace C136
{
    class Program
    {
        enum Apple { Jonathan, GoldenDel, RedDel, Winesap, Cortland,
        McIntosh };

        static void Main(string[] args)
        {
            string[] color = { "червоний", "жовтий", "червоний",
                " червоний", " червоний", " червонувато-зелений" };
            Apple i; // оголосити змінну переліченого типу
            // Використовувати змінну i для циклічного
            // звернення до членів переліку
            for (i = Apple.Jonathan; i <= Apple.McIntosh; i++)
                Console.WriteLine(i + " має значення " + (int)i);
            Console.WriteLine();
            // Використовувати переліки для індексування масиву
            for (i = Apple.Jonathan; i <= Apple.McIntosh; i++)
                Console.WriteLine("колір сорту " + i + "-" +
                    color[(int)i]);

            Console.ReadLine();
        }
    }
}
```

Результати розрахунку подано на рис. 9.4.



```
D:\Program\C136\C136\bin\Debug\C136.exe
Jonathan має значення 0
GoldenDel має значення 1
RedDel має значення 2
Winesap має значення 3
Cortland має значення 4
McIntosh має значення 5

колір сорту Jonathan-червоний
колір сорту GoldenDel-жовтий
колір сорту RedDel-червоний
колір сорту Winesap- червоний
колір сорту Cortland- червоний
колір сорту McIntosh- червонувато-зелений
```

Рисунок 9.4 – Результати розв’язання задачі 9.4

## 9.2 Завдання для виконання лабораторних робіт

**Задача 1.** Описати структуру з ім'ям STUDENT, що містить такі поля:

- прізвище й ініціали;
- номер групи;
- успішність (масив із п'яти елементів).

Написати програму, що виконує такі дії:

1) уведення із клавіатури даних у масив, що складається з десяти структур типу STUDENT (записи повинні бути впорядковані за зростанням номера групи);

2) виведення на екран прізвищ і номерів груп для всіх студентів, включених у масив, якщо середній бал студента більше 4,0 (якщо таких студентів немає, вивести відповідне повідомлення);

3) виведення на екран прізвищ і номерів груп для всіх студентів, включених у масив, які мають оцінки 4 і 5 (якщо таких студентів немає, вивести відповідне повідомлення);

4) виведення на екран прізвищ і номерів груп для всіх студентів, які мають хоча б одну оцінку 2 (якщо таких студентів немає, вивести відповідне повідомлення).

**Задача 2.** Описати структуру з ім'ям NOTE, що містить такі поля:

- прізвище й ініціали;
- номер телефону;
- дату народження (масив із трьох чисел).

Написати програму, що виконує такі дії:

1) уведення із клавіатури даних у масив, що складається з восьми елементів типу NOTE (записи повинні бути розміщені за алфавітом);

2) виведення на екран інформації про людину, прізвище якої уведено з клавіатури (якщо такої немає, вивести відповідне повідомлення).

**Задача 3.** Описати структуру з ім'ям WORKER, що містить такі поля:

- прізвище й ініціали працівника;
- назва займаної посади;
- рік ставання на роботу.

Написати програму, що виконує такі дії:

1) уведення із клавіатури даних у масив, що складається з десяти структур типу WORKER (записи повинні бути упорядковані за алфавітом);

2) виведення на екран прізвищ працівників, стаж роботи яких перевищує значення, уведене з клавіатури (якщо таких працівників немає, вивести відповідне повідомлення).

## Тема 10. ФАЙЛИ

### 10.1. Приклади розв'язання задач

**Задача 10.1.** Написати програму, яка

1. Створює файл масиву дійсних чисел:

ім'я файлу: "F:\\file1.dat";

кількість елементів масиву  $n = 15$ ;

елементи масиву обчислюється за формулою:

$$f_i = i^2 + i/15.0 - 12, \text{ де } i = 0, 1 \dots (n-1).$$

2. Зчитує дані з файлу "F:\\file1.dat", обчислює суму елементів файлу.

*Програмний код:*

```
using System;
using System.IO;

namespace C124
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 15; // кількість елементів масиву
            double[] F = new double[n];
            double[] FRES = new double[n];
            int i; double s;
            // Файл відкривається для запису
            FileStream fsW = new FileStream("F:\\file1.dat",
                FileMode.Create, FileAccess.Write);
            BinaryWriter bw = new BinaryWriter(fsW);
            Console.WriteLine("Обчислені елементи масиву");
            for (i = 0; i < n; i++)
            {
                F[i] = i * i + i / 15.0 - 12;
                // Обчислені елементи масиву виводяться на екран
                Console.WriteLine($" {F[i]:F3}");
                // Елементи масиву записуються у файл
                bw.Write(F[i]);
            }
            Console.WriteLine();
            bw.Close(); fsW.Close();
            s = 0;
            // Файл відкривається для читання
```

```

FileStream fsR = new FileStream("F:\\file1.dat",
                               FileMode.Open, FileAccess.Read);
BinaryReader br = new BinaryReader(fsR);
Console.WriteLine("Прочитані елементи файлу ");
for (i = 0; i < n; i++)
{
    // Елементи файлу вчитуються й записуються
    //в елементи масиву
    FRES[i] = br.ReadDouble();
    Console.WriteLine($" {FRES[i]:F3}");
    s = s + FRES[i];
}
br.Close(); fsR.Close();
Console.WriteLine($" \n s= {s:F3}");
Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 10.1.

```

D:\Program\C124\C124\bin\Debug\C124.exe
Обчислені елементи масиву
-12,000
-10,933
-7,867
-2,800
4,267
13,333
24,400
37,467
52,533
69,600
88,667
109,733
132,800
157,867
184,933
Прочитані елементи файлу
-12,000
-10,933
-7,867
-2,800
4,267
13,333
24,400
37,467
52,533
69,600
88,667
109,733
132,800
157,867
184,933
s = 842,000

```

Рисунок 10.1 – Результати розв’язання задачі 10.1

*Пояснення:*

1) Для того щоб використовувати в програмі файли, необхідно:

- а) підключити простір імен, у якому описуються стандартні класи для роботи з файлами (`using System.IO`);
- б) створити файловий потік і зв'язати його з фізичним файлом;
- в) виконати операції обміну (уведення-виведення);
- г) закрити файл.

Для роботи із числовими даними типу `int` або `double` краще створити об'єкти класів `BinaryWriter` і `BinaryReader`, методи яких `Write()` (перевантажений для різних типів даних), `ReadInt32()` і `ReadDouble()` якнайкраще підходять для цієї задачі.

```
2) FileStream fsW = new FileStream("F:\\file1.dat",  
                                   FileMode.Create, FileAccess.Write);  
BinaryWriter bw = new BinaryWriter(fsW);
```

Файл з ім'ям ("F:\\file1.dat") спочатку відкривається на запис, у результаті чого об'єктне посилання `fsW` типу `FileStream` набуває актуального значення. Поза цим посиланням будується об'єкт `bw` класу `BinaryWriter`:

```
BinaryWriter bw = new BinaryWriter(fsW);
```

При записі в тексті програми шляху до файлу слід пам'ятати, що символ `\` у рядкових константах позначає початок спеціальної послідовності символів, наприклад, `\n`. Тому при записі імені файлу символ `\` необхідно продублювати.

Якщо при відкритті файлу шлях до файлу не зазначений, то файл даних шукається або створюється в тому самому каталозі, де перебуває файл програми, що виконується:

```
D:\Program\C124\C124\bin\Debug\file1.dat
```

```
3) bw.Write(F[i]);
```

За допомогою перевантаженого методу `Write()` класу `BinaryWriter` у відкритий файл циклічно поелементно записуються елементи масиву.

```
4) bw.Close();  
   fsW.Close();
```

Об'єкт, що пише, і файл закриваються однойменними методами `Close()`.

```
5) FileStream fsR = new FileStream("F:\\file1.dat",  
                                   FileMode.Open, FileAccess.Read);  
BinaryReader br = new BinaryReader(fsR);
```

Файл "F:\\file1.dat" знову відкривається, але вже на читання.

6) `FRES[i] = br.ReadDouble();`

Вміст файлу "F:\\file1.dat" послідовно «вчитується» від імені об'єкта `br` типу `BinaryReader` методом `ReadDouble()` у масив-приймач `FRES`.

Для перевірки результату значення елементів масиву виводяться в консольне вікно програми за допомогою статичного методу `WriteLine()` бібліотечного класу `Console`:

`Console.WriteLine($" {FRES[i]:F3}");`

7) `br.Close();`

`fsR.Close();`

Об'єкт, що читає, і файл закриваються однойменними методами `Close()`.

У табл. 10.1 подано варіанти задач.

Таблиця 10.1 – Варіанти задач

Номер варіанту	Кількість елементів файлу (значення $n$ )	Формула, за якою обчислюються елементи файлу
1	$n = 12$	$f_i = 1,2i^3 - 7i - 94$ , де $i = 0,1,..(n-1)$
2	$n = 14$	$f_i = 2,5i^2 - 3i - 50$ , де $i = 0,1,..(n-1)$
3	$n = 15$	$f_i = (i-8)(i+5,7)$ , де $i = 0,1,..(n-1)$
4	$n = 12$	$f_i = (i-9,3)(3i+5)$ , де $i = 0,1,..(n-1)$
5	$n = 15$	$f_i = (i-10)(5i+9)$ , де $i = 0,1,..(n-1)$
6	$n = 14$	$f_i = 2i^3 - 9i - 100$ , де $i = 0,1,..(n-1)$
7	$n = 12$	$f_i = 1,7i^3 - 12i - 70$ , де $i = 0,1,..(n-1)$
8	$n = 15$	$f_i = 14i^2 - 105i + 60$ , де $i = 0,1,..(n-1)$
9	$n = 12$	$f_i = 9i^2 - 87i + 60$ , де $i = 0,1,..(n-1)$
10	$n = 15$	$f_i = i^2 + 15i - 90$ , де $i = 0,1,..(n-1)$

**Задача 10.2.** Написати програму, яка зчитує дані з файлу ("F:\\file1.dat"), створеного в задачі 11.1. Отримати два нових файли з іменами ("F:\\file2.dat") і ("F:\\file3.dat"), в один з яких записати додатні елементи файлу, а в другий – від'ємні. Обчислити кількість елементів отриманих файлів. Отримані файли прочитати і вивести на екран.

*Програмний код:*

```
using System;
using System.IO;

namespace C128
{
    class Program
    {
        static void Main(string[] args)
        {
            const int n = 15; // кількість елементів масиву
            double[] FRES = new double[n];
            int i;
            int n1; // кількість елементів файлу file2.dat
            int n2; // кількість елементів файлу file3.dat
            // Файл відкривається для читання
            FileStream fsR = new FileStream("F:\\file1.dat",
                FileMode.Open, FileAccess.Read);
            BinaryReader br = new BinaryReader(fsR);
            // Файл відкривається для запису
            FileStream fsW1 = new FileStream("F:\\file2.dat",
                FileMode.Create, FileAccess.Write);
            BinaryWriter bw1 = new BinaryWriter(fsW1);
            // Файл відкривається для запису
            FileStream fsW2 = new FileStream("F:\\file3.dat",
                FileMode.Create, FileAccess.Write);
            BinaryWriter bw2 = new BinaryWriter(fsW2);
            n1 = 0; n2 = 0;
            Console.WriteLine("Елементи файлу file1.dat ");
            for (i = 0; i < n; i++)
            {
                // Читання файлу
                FRES[i] = br.ReadDouble();
                // Виведення елементів файлу на екран
                Console.WriteLine($" {FRES[i]:F3}");
                if (FRES[i] >= 0)
                {
                    bw1.Write(FRES[i]);
                    n1 = n1 + 1;
                }
                else
                {
                    bw2.Write(FRES[i]);
                }
            }
        }
    }
}
```

```

        n2 = n2 + 1;
    }
}
br.Close(); fsR.Close(); bw1.Close();
fsW1.Close(); bw2.Close(); fsW2.Close();
Console.WriteLine();
Console.WriteLine("n1={0}; n2={1}", n1, n2);
Console.WriteLine();
// Файл відкривається для читання
FileStream fsR2 = new FileStream("F:\\file2.dat",
    FileMode.Open, FileAccess.Read);
BinaryReader br2 = new BinaryReader(fsR2);
// Файл відкривається для читання
FileStream fsR3 = new FileStream("F:\\file3.dat",
    FileMode.Open, FileAccess.Read);
BinaryReader br3 = new BinaryReader(fsR3);
double[] F2 = new double[n1];
double[] F3 = new double[n2];
Console.WriteLine("Елементи файлу file2.dat ");
for (i = 0; i < n1; i++)
{
    // Читання файлу
    F2[i] = br2.ReadDouble();
    // Виведення елементів файлу на екран
    Console.WriteLine($" {F2[i]:F3}");
}
Console.WriteLine();
Console.WriteLine("Елементи файлу file3.dat \n");
for (i = 0; i < n2; i++)
{
    // Читання файлу
    F3[i] = br3.ReadDouble();
    // Виведення елементів файлу на екран
    Console.WriteLine($" {F3[i]:F3}");
}
br2.Close();
fsR2.Close();
br3.Close();
fsR3.Close();
Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 10.2.

```
D:\Program\C128\C128\bin\Debug\C128.exe
Елементи файлу file1.dat
-12,000
-10,933
-7,867
-2,800
4,267
13,333
24,400
37,467
52,533
69,600
88,667
109,733
132,800
157,867
184,933
n1=11; n2=4
Елементи файлу file2.dat
4,267
13,333
24,400
37,467
52,533
69,600
88,667
109,733
132,800
157,867
184,933
Елементи файлу file3.dat
-12,000
-10,933
-7,867
-2,800
```

Рисунок 10.2 – Результати розв’язання задачі 10.2

**Задача 10.3.** Створити текстовий файл, у який буде виведена таблиця значень функції  $y = 5x + 3\sin(x) + 7$  в діапазоні від  $1 \leq x \leq 10$  з кроком 1. Таблиця має складатися з двох стовпців: значень аргументу і відповідних їм значень функції.

*Програмний код:*

```
using System;
using System.IO;

namespace C123
{
    class Program
    {
```

```

static void Main(string[] args)
{
    // Файл відкривається для запису
    FileStream fsW = new FileStream("F:\\file4.txt",
        FileMode.Create, FileAccess.Write);
    StreamWriter sw = new StreamWriter(fsW);
    double x, y;          // аргумент і значення функції
    double xn = 1.0;     // початкове значення змінної x
    double xk = 10.0;   // кінцеве значення змінної x
    double dx = 1.0;    // крок зміни змінної
    string s;
    Console.WriteLine("Обчислені значення функції у ");
    for (x = xn; x <= xk; x = x + dx)
    {
        y = 5 * x + 3 * Math.Sin(x) + 7;
        // Обчислені значення у виводяться на екран
        Console.WriteLine($"x={x,6:F2} y={y,8:F3}");
        // Обчислені значення у записуються до файлу
        s = $"x={x,6:F2} y={y,8:F3}";
        sw.WriteLine(s);
    }
    Console.WriteLine();
    sw.Close();
    fsW.Close();
    // Файл відкривається для читання
    FileStream fsR = new FileStream("F:\\file4.txt",
        FileMode.Open, FileAccess.Read);
    StreamReader sr = new StreamReader(fsR);
    Console.WriteLine("Прочитані елементи файлу");
    // Цикл виконується доти, доки не досягнуто кінця файлу
    while ((s = sr.ReadLine()) != null)
        Console.WriteLine(s);
    sr.Close();
    fsR.Close();
    Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 10.3.

```
D:\Program\C123\C123\bin\Debug\C123.exe
Обчислені значення функції у
x= 1,00 y= 14,524
x= 2,00 y= 19,728
x= 3,00 y= 22,423
x= 4,00 y= 24,730
x= 5,00 y= 29,123
x= 6,00 y= 36,162
x= 7,00 y= 43,971
x= 8,00 y= 49,968
x= 9,00 y= 53,236
x= 10,00 y= 55,368

Прочитані елементи файлу
x= 1,00 y= 14,524
x= 2,00 y= 19,728
x= 3,00 y= 22,423
x= 4,00 y= 24,730
x= 5,00 y= 29,123
x= 6,00 y= 36,162
x= 7,00 y= 43,971
x= 8,00 y= 49,968
x= 9,00 y= 53,236
x= 10,00 y= 55,368
```

Рисунок 10.3 – Результати розв’язання задачі 10.3

*Пояснення:*

З текстовими файлами краще працювати за допомогою об’єктів спеціалізованих класів `StreamWriter` і `StreamReader`, які пишуть і читають. Тут використовуються методи `Write()` і `ReadLine()`.

Програма за один виклик методу `ReadLine()` читає один фізичний рядок. Умова повного вичитування текстового файлу – повертання методом `ReadLine()` значення `null`.

Запис рядків до текстового файлу краще здійснювати не методом `Write()`, а методом `WriteLine()`.

**Задача 10.4.** Є текстовий файл (слова розділено одним пробілом). Прочитати файл. Визначити, скільки в тексті голосних та приголосних літер. Зміст текстового файлу ("F:\file5.txt"):

**ВСТАЛА ВЕСНА**

Встала весна, чорну землю  
Сонну розбудила,  
Уквітчала її рястом,  
Барвінком укрила:  
І на полі жайворонок,  
Соловейко в гаї  
Землю, убрану весною,  
Вранці зустрічають.

*Тарас Шевченко*

*Програмний код:*

```
using System;
using System.IO;

namespace C125
{
    class Program
    {
        static void Main(string[] args)
        {
            char[] c1 = { 'a', 'A', 'e', 'E', 'и', 'И', 'і', 'І',
                        'o', 'O', 'y', 'Y', 'є', 'Є', 'ю', 'Ю',
                        'я', 'Я', 'ї', 'Ї'};
            char[] c2 = { 'б', 'Б', 'в', 'В', 'г', 'Г', 'д', 'Д',
                        'ж', 'Ж', 'з', 'З', 'й', 'Й', 'к', 'К', 'л', 'Л',
                        'м', 'М', 'н', 'Н', 'п', 'П', 'р', 'Р',
                        'с', 'С', 'т', 'Т', 'ф', 'Ф', 'х', 'Х',
                        'ц', 'Ц', 'ч', 'Ч', 'ш', 'Ш', 'щ', 'Щ'};
            int n1 = 0; // кількість голосних букв
            int n2 = 0; // кількість приголосних букв
            string Str;
            // Файл відкривається для читання
            FileStream fsR = new FileStream("F:\\file5.txt",
                                           FileMode.Open, FileAccess.Read);
            StreamReader sr = new StreamReader(fsR);
            while ((Str = sr.ReadLine()) != null)
            {
                Console.WriteLine(Str);
                // Обчислення кількості голосних букв
                for (int i = 0; i < Str.Length; i++)
                {
                    char c0 = Str[i];
                    for (int j = 0; j < c1.Length; j++)
                        if (c0 == c1[j])
                            n1 = n1 + 1;
                    for (int j = 0; j < c2.Length; j++)
                        if (c0 == c2[j])
                            n2 = n2 + 1;
                }
            }
            sr.Close();
            fsR.Close();
            Console.WriteLine();
        }
    }
}
```

```

        Console.WriteLine("Кількість голосних символів: " + n1);
        Console.WriteLine("Кількість приголосних символів: " + n2);
        Console.ReadLine();
    }
}

```

Результати розрахунку подано на рис. 10.4.

```

D:\Program\C125\C125\bin\Debug\C125.exe
ВСТАЛА ВЕСНА
Встала весна, чорну землю
Сонну розбудила,
Уквітчала її рястом,
Барвінком укрила:
І на полі жайворонок,
Соловейко в гаї
Землю, убрану весною,
Вранці зустрічають.
Тарас Шевченко
Кількість голосних символів: 59
Кількість приголосних символів: 89

```

Рисунок 10.4 – Результати розв’язання задачі 10.4

*Пояснення:* Задано символний масив `s1`, який містить голосні букви, символний масив `s2`, який містить приголосні букви. Рядок `Str`, прочитаний з файлу, перебирається посимвольно в циклі до довжини рядка, звертаючись до елементів рядка за індексом.

Кожний символ у вкладеному циклі по черзі порівнюється з елементами масиву голосних букв. У випадку порівняння значення лічильника `n1` збільшується на одиницю. Кожний символ у вкладеному циклі по черзі порівнюється також з елементами масиву приголосних букв. У випадку порівняння значення лічильника `n2` збільшується на одиницю.

**Задача 10.5.** Створити квадратну матрицю  $A$  розміром  $n \times n$ .

Матрицю заповнити випадковими цілими числами.

Матрицю по рядках вивести на екран.

У текстовий файл "F:\file10.txt" записати спочатку  $n$  – кількість рядків і стовпців матриці, а потім – саму матрицю.

Прочитати отриманий файл.

*Програмний код:*

```

using System;
using System.IO;

namespace C112
{
    class Program
    {

```

```

static void Main(string[] args)
{
    int n; // кількість рядків і стовпців матриці A
    Console.Write("Введіть n= ");
    n = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine();
    int i, j, k;
    double[,] A = new double[n, n];
    string S;
    Random rnd = new Random();
    // Файл відкривається для запису
    FileStream fsW = new FileStream("F:\\file10.txt",
        FileMode.Create, FileAccess.Write);
    StreamWriter sw = new StreamWriter(fsW);
    // У файл записується спочатку розмір матриці n,
    // далі сама матриця за рядками
    Console.WriteLine($"n={n}");
    sw.WriteLine(n);
    // Заповнення матриці A,
    // виведення матриці A на екран і запис до файлу
    Console.WriteLine("Елементи матриці A ");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            k = rnd.Next(1, 9);
            Console.Write(k + " ");
            sw.Write(k + " ");
        }
        Console.WriteLine(); sw.WriteLine();
    }
    Console.WriteLine();
    sw.Close(); fsW.Close();
    // Файл відкривається для читання
    FileStream fsR = new FileStream("F:\\file10.txt",
        FileMode.Open, FileAccess.Read);
    StreamReader sr = new StreamReader(fsR);
    S = sr.ReadLine(); // Прочитати рядок
    n = Convert.ToInt32(S); // Визначити розмір матриці
    Console.WriteLine($"n= {n}");
    Console.WriteLine("Елементи файлу file10.dat ");
    // Читання файлу і виведення на екран

```



У переліку `StringSplitOptions` визначаються тільки два значення: `None` і `RemoveEmptyEntries`. При значенні `None` порожні рядки включаються в кінцевий результат розподілу цього рядка. При значенні `RemoveEmptyEntries` порожні рядки виключаються з кінцевого результату поділу цього рядка.

**Задача 10.6.** Створити квадратну матрицю  $A$  розміром  $n \times n$ . Кількість рядків  $n$  і самі елементи матриці ввести з файлу ("F:\\file8.txt"). Одержати нову матрицю  $B$  розміром  $n \times n$ . Нова матриця буде дорівнювати матриці  $A$ , в якій потрібно замінити додатні елементи їх кубами, а інші – подвоїти. Виведення результатів здійснити у файл ("F:\\file9.txt"), спочатку вивести кількість рядків  $n$ , а потім – самі елементи матриці  $B$ .

Файл ("F:\\file8.txt")

```
4
1,5
3,4
-5,8
8,0
-10,2
12,4
14,8
-15,5
17,3
20,4
-21,9
22,4
24,2
-26,3
27,5
30,8
```

*Програмний код:*

```
using System;
using System.IO;

namespace C111
{
    class Program
    {
        static void Main(string[] args)
        {
            int i, j;
            double c;
            string S;
            int n; // кількість рядків і стовпців матриці A
            // Файл відкривається для читання
```

```

FileStream fsR = new FileStream("F:\\file8.txt",
                               FileMode.Open, FileAccess.Read);
StreamReader sr = new StreamReader(fsR);
Console.WriteLine("Елементи файлу file8.dat - матриця A");
// Читання файлу і виведення на екран
// З файлу зчитується спочатку розмір матриці n,
// далі сама матриця по рядках
S = sr.ReadLine(); //прочитати рядок
n = Convert.ToInt32(S); //визначити розмір матриці
Console.WriteLine($"n = {n}");
double[,] A = new double[n, n];
double[,] B = new double[n, n];
for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
        c = Convert.ToDouble(sr.ReadLine());
        A[i, j] = c;
        Console.Write($"{A[i,j],9:F2} ");
    }
    Console.WriteLine();
}
Console.WriteLine();
sr.Close();
fsR.Close();
// Обчислення нової матриці B
for (i = 0; i < n; i++)
{
    for (j = 0; j < n; j++)
    {
        if (A[i, j] > 0)
            B[i, j] = Math.Pow(A[i, j], 3);
        else
            B[i, j] = 2 * A[i, j];
    }
}
Console.WriteLine();
// Файл відкривається для запису
FileStream fsW = new FileStream("F:\\file9.txt",
                               FileMode.Create, FileAccess.Write);
StreamWriter sw = new StreamWriter(fsW);
// До файлу записується спочатку розмір матриці n,
// далі сама матриця за рядками
Console.WriteLine($"n={n}");
sw.WriteLine(n);
// Виведення нової матриці B на екран і запис до файлу
Console.WriteLine("Елементи матриці B ");

```

```

    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
        {
            sw.WriteLine(B[i, j]);
            Console.WriteLine($"{B[i,j],9:F3}  ");
        }
        Console.WriteLine();
    }
    Console.WriteLine();
    sw.Close();
    fsW.Close();
    Console.ReadLine();
}
}
}

```

Результати розрахунку подано на рис. 10.6.

```

D:\Program\C111\C111\bin\Debug\C111.exe
Елементи файлу file8.dat - матриця А
n = 4
  1,50      3,40     -5,80      8,00
 -10,20    12,40     14,80    -15,50
  17,30    20,40    -21,90     22,40
  24,20   -26,30     27,50     30,80

n=4
Елементи матриці В
  3,375    39,304    -11,600    512,000
 -20,400  1906,624  3241,792   -31,000
 5177,717  8489,664   -43,800  11239,424
14172,488  -52,600  20796,875  29218,112

```

Рисунок 10.6 – Результати розв'язання задачі 10.6

Файл ("F:\\file9.txt")

```

4
3,375
39,304
-11,6
512
-20,4
1906,624
3241,792
-31
5177,717
8489,664
-43,8
11239,424
14172,488
-52,6
20796,875
29218,112

```

Навчальне видання

**СОЛОВЕЙ Людмила Валентинівна**  
**МІРОШНІЧЕНКО Наталія Миколаївна**  
**БАБАК Тетяна Геннадіївна**

## **ПРОГРАМУВАННЯ МОВОЮ C# Visual Studio 2019**

### **Лабораторний практикум**

для студентів хімічних спеціальностей усіх форм навчання

У трьох частинах

### **Частина 2**

Відповідальний за випуск проф. Ведь В.Є.

Роботу до видання рекомендував проф. Рисований О.М.

В авторській редакції

План 2021 р., поз. 38

Підписано до друку 22.02.2022 р.    Формат 60×84 1/16.    Папір офсетний.

Riso-друк.    Гарнітура Таймс.    Ум. друк. арк.    Наклад 50 прим.

Зам №    Ціна договірна

---

Видавничий центр НТУ «ХП», вул. Кирпичова, 2, м. Харків-2, 61002

Свідоцтво суб'єкта видавничої справи ДК №5478 від 21.08.2017 р.

---