

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДВНЗ «Ужгородський національний університет»
Факультет інформаційних технологій
Кафедра програмного забезпечення систем

В. М. Коцовський

Методичні матеріали до лабораторних робіт

з дисципліни

«Технологія програмування та створення програмних продуктів»

Ужгород – 2016

Лабораторна робота № 1

Теми:

1. Обробка масивів у C#.
2. Форми та візуальні елементи керування .NET.

Вимоги до роботи: завдання мають бути виконанні з використанням Windows Forms. Елементи масивів потрібно зобразити на формі. Частина коду необхідно реалізувати у вигляді функцій користувача.

Приклад. Написати програму для знаходження усіх елементів масиву, які мають найбільшу суму цифр серед чисел, які потрапляють у задані межі.

Текст основного модуля:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Windows.Forms;

namespace ArraySample
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private int SumOfDigits(int n)
        {
            int s = 0;
            while (n > 0)
            {
                s += n%10;
                n /= 10;
            }
            return s;
        }

        private void addButton_Click(object sender, EventArgs e)
        {
            int number;
            if (int.TryParse(itemTextBox.Text, out number) && number >= 0)
                arrayListBox.Items.Add(number);
            else
            {
                itemTextBox.Select();
                itemTextBox.Select(0, itemTextBox.Text.Length);
            }
        }
    }
}
```

```

}

private void removeButton_Click(object sender, EventArgs e)
{
    if(arrayListBox.SelectedIndex >= 0)
        arrayListBox.Items.RemoveAt(arrayListBox.SelectedIndex);
}

private void runButton_Click(object sender, EventArgs e)
{
    int lower = int.Parse(lowerTextBox.Text);
    int upper = int.Parse(uperTextBox.Text);
    int[] array = new int[arrayListBox.Items.Count];
    int maxSum = int.MinValue;
    List<int> numbers = new List<int>();
    for (int i = 0; i < array.Length; i++)
    {
        array[i] = Convert.ToInt32(arrayListBox.Items[i]);
        if (array[i] >= lower && array[i] <= upper)
        {
            int sum = SumOfDigits(array[i]);
            if(sum == maxSum)
                numbers.Add(array[i]);
            if (sum > maxSum)
            {
                maxSum = sum;
                numbers.Clear();
                numbers.Add(array[i]);
            }
        }
    }
    if(numbers.Count > 0)
        MessageBox.Show(string.Format("Numbers with max digits
sum:\n {0}", string.Join("\n", numbers)), "Result");
    else
    {
        MessageBox.Show("No numbers found", "Result");
    }
}
}
}

```

Вікно програми та приклад роботи програми наведений на рис. 1.

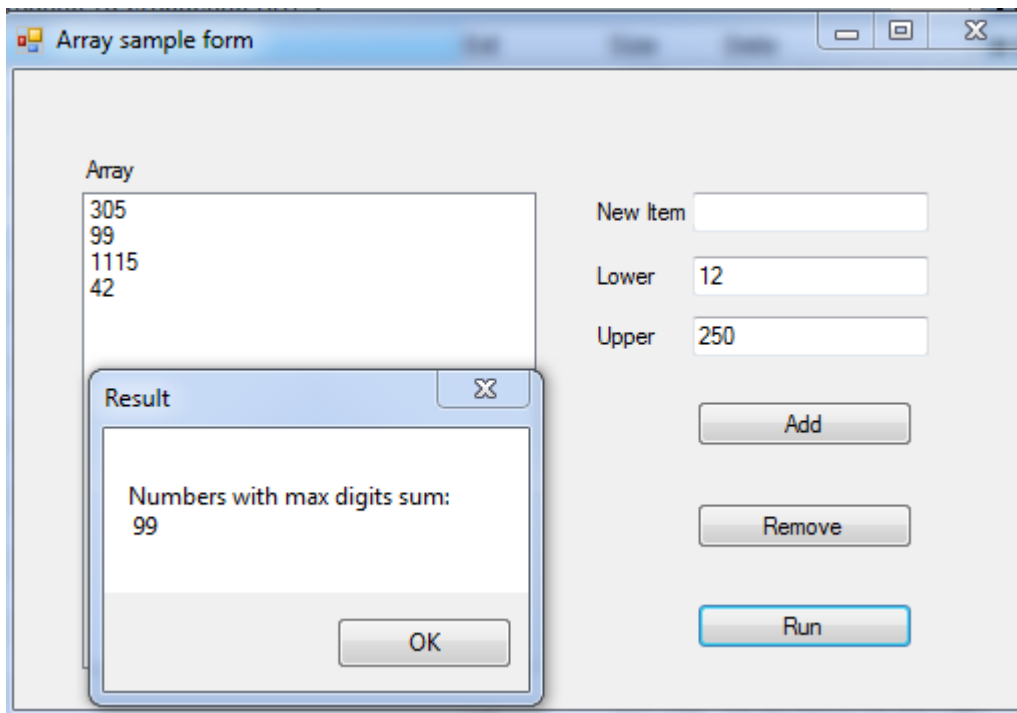


Рис. 1. Вікно програми

Варіанти завдань

1. Користувач вводить кількість елементів та самі елементи одновимірного масиву рядків. Вивести індекси елементів, які можуть бути конвертовані у числовий тип і мають найменшу дробову частину.
2. Користувач вводить кількість елементів та самі елементи одновимірного числового масиву. Поміняти місцями найбільший від'ємний та найменший додатний елементи масиву.
3. Елементи масиву $A = (a_i)$ задаються так:

$$a_1 = -4, a_2 = 3, a_i = a_{i-1}^2 + 2a_{i-2} - i, i = 3, 4, \dots, n.$$

Знайти середнє арифметичне всіх елементів масиву, які потрапляють у проміжок $(b, c]$. Всі інші елементи записати у окремий ListBox, впорядкувавши їх в порядку спадання.

4. Користувач вводить кількість елементів та самі елементи одновимірного масиву рядків. Вивести ті елементи масиву, які можуть бути записані у вигляді конкатенації кількох своїх безпосередніх попередників. Наприклад, для масиву {"news", "maker", "newsmaker", "step", "by", "step", "stepbystep"} шуканими будуть "newsmaker" та "stepbystep".
5. Користувач вводить кількість елементів та самі елементи одновимірного цілочислового масиву. Вивести ті елементи, які не діляться націло на добуток двох яких-небудь інших елементів масиву.
6. На формі вводяться числа a_i , b_i та c_i ($i = \overline{1, n}$). Вивести ті трійки чисел, які утворюють трикутники найбільшого периметру.
7. Користувач задає масив чисел. Вивести середнє арифметичне тих парних елементів масиву, які не перевищують добуток цифр свого індексу.

8. Знайти добуток елементів числового масиву $A = (a_i), i = \overline{1, n}$, які потрапляють у проміжок $[bt, cM]$, де t — найменший елемент масиву A , M — найбільший елемент масиву (числа b та c вводяться у елементи керування на формі).
9. На формі вводяться натуральні числа $a_i (i = \overline{1, n})$. Вивести ті числа, які утворені лише із непарних цифр.
10. Користувач вводить n рядків та символ c . Вивести ті рядки, індекс першого входження у які символу c найбільший.
11. Запрограмувати зсув елементів одновимірного масиву на заданому кількості позицій у вказаному напрямку (вліво на вправо)
- циклічно.
 - із очисткою звільнених елементів.

12. Користувач задає елементи масиву $B = (b_i)$. Сформувати масив $A = (a_i)$ таким чином:

$$a_1 = b_1 \sin 1, \quad a_2 = b_1 \sin 1 - b_2 \cos^2 3, \quad a_3 = b_1 \sin 1 - b_2 \cos^2 3 + b_3 \sin^3 5, \dots$$

Вивести усі елементи масиву A , які менші за середнє арифметичне його додатних елементів.

13. Впорядкувати елементи цілочислового масиву у порядку зростання суми їх цифр.
14. Користувач задає масив рядків. Вивести ті елементи масиву, довжина яких більша за довжину сусідніх елементів.
15. Користувач задає 1-ий масив натуральних чисел. На основі цього масиву обчислюється 2-ий масив, кожний елемент якого рівний сумі цифр відповідного елемента 1-го масиву. За тим самим принципом обчислюються 3-ий, 4-ий, ... масиви до тих пір, поки новий масив не буде співпадати із своїм попередником. Вивести цей масив.
16. Елементи масиву $A = (a_i), (i = 1, 2, \dots, n)$ (x та n задаються користувачем) задаються так:

$$a_1 = e^{\sin x}, \quad a_2 = e^{\sin x - \sin 2x}, \quad a_3 = e^{\sin x - \sin 2x + \sin 3x}, \dots$$

Замінити на 0 ті елементи масиву, які не належать проміжку $[b, c)$. Вивести змінений масив.

17. Підрахувати кількість елементів одновимірного числового масиву, модуль яких не менший за p відсотків максимального елемента.
18. Користувач задає масив рядків. Вивести кількість рядків, які утворені із різних символів (жодний символ у яких не повторюється).
19. На формі вводяться числа a_i та $b_i (i = \overline{1, n})$. Вивести число, яке найчастіше зустрічається серед добутоків $a_i b_i$.
20. Користувач задає масив рядків. Вивести список рядків, які містять найбільшу кількість пробілів.
21. Користувач задає масив чисел. Вивести довжину найдовшої спадної послідовності елементів масиву.
22. Елементи масиву $A = (a_i), (i = 1, 2, \dots, n)$ задаються так:

$$a_i = \frac{\cos 1 \cdot (3 \cos 2) \cdot (5 \cos 3) \cdot \dots \cdot ((2i-1) \cos i)}{1 + 4 + 9 + \dots + i^2} b_i,$$

де елементи масиву $B = (b_i)$ вводить користувач. Записати у ComboBox від'ємні елементи масиву A , впорядковані в порядку зростання.

23. На формі вводяться координати точок A_i, B_i, C_i ($i = \overline{1, n}$). Вивести координати вершин трикутників $A_i B_i C_i$, площа яких найбільша.
24. На формі вводяться дійсні невід'ємні числа x_i ($i = \overline{1, n}$). Вивести ті числа, дробова частина яких більша за p відсотків від цілої частини (p вводить користувач).
25. Користувач вводить n рядків. Вивести ті рядки, які містять найменшу кількість великих латинських літер.
26. Елементи масиву $A = (a_i)$ задаються на формі. Вивести найдовшу послідовність a_k, a_{k+1}, \dots, a_m елементів масиву A , які задовольняють умову $a_i < |a_{i+1}|$, $i = k, k+1, \dots, m-1$.
27. Знайти суму індексів парних цілих елементів числового масиву.
28. Елементи масиву $A = (a_i)$ задаються на формі. Побудувати масив $B = (b_i)$ наступним чином:

$$b_i = \begin{cases} a_1 \cdot a_2 \cdot \dots \cdot a_i, & \text{якщо } a_i < 0, \\ |a_1| + 2|a_2| + \dots + i|a_i|, & \text{в іншому випадку,} \end{cases}$$

та вивести його елементи у зворотному порядку.

29. На формі вводяться числа a_i та b_i — довжини сторін прямокутників ($i = \overline{1, n}$). Вивести ті прямокутники, площа яких рівна найменшій із площ квадратів, які зустрічаються серед заданих прямокутників.
30. Користувач задає масив рядків. Вивести довжину найдовшої послідовності однакових елементів масиву (елементи послідовності йдуть один за одним).
31. На формі вводяться натуральні числа a_i ($i = \overline{1, n}$). Вивести ті числа, у десятковому записі яких не менше половини цифр — нулі.
32. Користувач вводить масив рядків. Вивести елементи масиву, які можна перетворити у число і які найменше відрізняються від свого індексу.
33. Користувач вводить n рядків та рядок s . Вивести ті рядки, у які входять усі літери рядка s .
34. Елементи числового масиву $A = (a_i)$, $i = \overline{0, n}$ вводяться користувачем. Знайти $[a_1 - a_2]! + \dots + [a_{n-1} - a_n]!$, де $[x]$ — ціла частина числа x , $y!$ — факторіал невід'ємного числа y .
35. Вивести натуральні елементи числового масиву, індекс яких більший за суму їх цифр, та знайти середнє арифметичне цих елементів.
36. Користувач задає масив чисел. Вивести добуток тих елементів, кожний з яких менший за усі попередні елементи масиву.
37. Вивести усі спільні елементи трьох заданих рядкових масивів крім елементів, які мають найменшу довжину серед спільних елементів масивів.

38. Користувач задає масив чисел. Вивести добуток тих з них, які є цілими додатними числами та утворені лише з парних цифр.
39. На формі вводяться натуральні числа a_i ($i = \overline{1, n}$). Вивести ті числа, усі цифри десяткового запису яких різні.
40. Елементи масиву $A = (a_i)$, задаються так:
- $$a_1 = 1, a_i = \sin b + \sin \frac{b}{2} + \dots + \sin \frac{b}{2^{i-2}}, i = 2, \dots, n.$$
- Вивести k найменших елементів масиву A .
41. Користувач задає масив рядків. Для кожного рядка вивести суму усіх чисел, які входять у цей рядок.
42. Користувач вводить кількість елементів та самі елементи одновимірного числового масиву. Вивести суму елементів, які мають найменшу дробову частину.
43. Користувач вводить кількість елементів та самі елементи одновимірного числового масиву. Обнулити ті елементи масиву, які розташовані між найбільшим та найменшим елементами.
44. Користувач вводить елементи одновимірного цілочислового масиву. Вивести ті елементи, які діляться націло принаймні на половину з усіх елементів масиву.
45. На формі вводяться числа a_i, b_i та c_i ($i = \overline{1, n}$). Вивести ті трійки чисел, які утворюють трикутники найменшої площі.
46. На формі вводяться натуральні числа a_i ($i = \overline{1, n}$). Вивести ті числа, у десятковому записі яких зустрічаються усі 10 цифр.
47. Користувач задає масив рядків. Вивести ті елементи масиву, довжина яких менша за суму довжин сусідніх елементів.
48. Впорядкувати елементи цілочислового масиву у порядку спадання добутку їх цифр.
49. Користувач вводить кількість елементів та самі елементи одновимірного числового масиву. Якщо найменший елемент масиву розташований перед найбільшим, то обнулити ті елементи масиву, які розташовані перед найменшим елементом, інакше ті елементи, які розташовані за найменшим.
50. На формі вводяться числа a_i та b_i — довжини сторін прямокутників ($i = \overline{1, n}$). Вивести середню площу тих прямокутників, периметр яких найбільший.
51. Користувач задає масив чисел. Вивести найдовші арифметичні прогресії, утворені послідовними елементами масиву. Наприклад, для масиву 1, 3, 6, 9, 8, 7 результатом будуть 2 прогресії 3, 6, 9 та 9, 8, 7.
52. Користувач задає масив чисел. Вивести ті елементи, які менші за принаймні які-небудь два попередні елементи масиву.
53. На формі вводяться дійсні числа x_i ($i = \overline{1, n}$). Вивести ті числа, дробова частина яких менша за p відсотків від максимальної цілої частини усіх чисел (p вводить користувач).
54. На формі вводяться числа a_i, b_i та c_i ($i = \overline{1, n}$). Вивести ті трійки чисел, які утворюють геометричні прогресії з найбільшим знаменником.
55. На формі вводяться числа a_i, b_i, c_i та d_i ($i = \overline{1, n}$). Вивести ті четвірки чисел, які утворюють арифметичні прогресії з найменшою різницею.

56. Користувач задає масив чисел. Вивести ті елементи, кожний з яких більший за усі наступні елементи масиву.
57. На формі вводяться натуральні числа a_i ($i = \overline{1, n}$). Вивести ті числа, у десятковому записі яких зустрічаються не більше трьох різних цифр.
58. Знайти добуток індексів цілих тих елементів числового масиву, які кратні a та b (числа a та b вводить користувач).
59. Користувач вводить кількість елементів та самі елементи одновимірного масиву рядків. Вивести ті елементи масиву, які можуть бути записані у вигляді конкатенації кількох своїх безпосередніх наступників. Наприклад, для масиву {"warmonger", "war", "monger", "single", "single", } шуканими будуть "warmonger" та "single".
60. Користувач вводить n рядків та символ s . Вивести ті рядки, індекс останнього входження у які символу s найменший.

Лабораторна робота № 2

Теми:

1. Обробка матриць у C#.
2. Файлові потоки.
3. Класи та компоненти для роботи з табличними та файловими даними в .NET.
4. Засоби для створення ділової графіки.

Вимоги до роботи: завдання мають бути виконанні з використанням Windows Forms. Елементи матриць потрібно зобразити на формі. Потрібно передбачити можливість запису та зчитування вмісту файлу на форму з використанням файлових потоків. При вказівці імен файлів даних використати файлові діалоги. Частина коду необхідно реалізувати у вигляді функцій користувача. Результат вивести на окрему форму.

Приклад. Користувач вказує два файли, які містять елементи цілочислових матриць. Розв'язати наступні завдання:

1. Виділити кольором ті стовпці першої матриці, кожний елемент яких більший ніж відповідний йому елемент другої матриці, розташований у тій самій позиції.
2. Зобразити гістограму, поставивши кожній матриці у відповідність окремий ряд даних. На осі абсцис треба відкласти натуральні числа від 1 до M , а на осі ординат — кількість елементів матриці, до складу яких входить не менше, ніж k цифр, що повторюються, де k — абсциса точки графіку, M — максимальна кількість цифр, що повторюються.

Текст 1-го модуля:

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Windows.Forms;
using System.IO;

namespace L2Sample
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void task1Button_Click(object sender, EventArgs e)
        {
            int n1, n2;
```

```

for (int j = 0; j < dataGridView1.ColumnCount; j++)
{
    bool needToSelect = true;
    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        if (int.TryParse(dataGridView1[j, i].Value.ToString(),
out n1) &&
int.TryParse(dataGridView2[j, i].Value.ToString(),
out n2))
        {
            if (n1 <= n2)
                needToSelect = false;
        }
        else
        {
            MessageBox.Show("Invalid non integer data");
            return;
        }
    }
    if (needToSelect)
    {
        dataGridView1.Columns[j].DefaultCellStyle.Font = new
Font(dataGridView1.DefaultCellStyle.Font,
        FontStyle.Underline);
        dataGridView1.Columns[j].DefaultCellStyle.BackColor =
Color.BurlyWood;
    }
    else
    {
        dataGridView1.Columns[j].DefaultCellStyle.Font = new
Font(dataGridView1.DefaultCellStyle.Font,
        FontStyle.Regular);
        dataGridView1.Columns[j].DefaultCellStyle.BackColor =
Color.White;
    }
}
}

void ReadMatrix(DataGridView grid)
{
    openFileDialog1.InitialDirectory = Application.StartupPath;
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        grid.Rows.Clear();
        using (var reader = new
            StreamReader(openFileDialog1.FileName))
        {
            var words = reader.ReadLine().Split(new[] { ' ', '\t' },
                StringSplitOptions.RemoveEmptyEntries);
            grid.ColumnCount = words.Length;
            grid.Rows.Add(words.ToArray());
            while (!reader.EndOfStream)
            {
                words = reader.ReadLine().Split(new[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);
                grid.Rows.Add(words);
            }
        }
    }
}

```

```

        }
    }
}

private void readFirstButton_Click(object sender, EventArgs e)
{
    ReadMatrix(dataGridView1);
    task1Button.Enabled = dataGridView1.RowCount ==
        dataGridView2.RowCount && dataGridView1.ColumnCount ==
            dataGridView2.ColumnCount;
}

private void readSecondButton_Click(object sender, EventArgs e)
{
    ReadMatrix(dataGridView2);
    task1Button.Enabled = dataGridView1.RowCount ==
        dataGridView2.RowCount && dataGridView1.ColumnCount ==
            dataGridView2.ColumnCount;
}

int MultipleDigitCount(int n)
{
    var digits = new HashSet<int>();
    var multipleDigits = new HashSet<int>();
    while (n > 0)
    {
        int d = n % 10;
        if (digits.Contains(d))
            multipleDigits.Add(d);
        else
            digits.Add(d);
        n /= 10;
    }
    return multipleDigits.Count;
}

private int[] GetNumbers(DataGridView grid)
{
    var matrix = new int[grid.RowCount, grid.ColumnCount];
    var result = new int[10];
    for (int i = 0; i < grid.RowCount; i++)
    {
        for (int j = 0; j < grid.ColumnCount; j++)
        {
            if (!int.TryParse(grid[j, i].Value.ToString(), out
matrix[i, j]))
            {
                MessageBox.Show("Invalid non integer data", "Data
                Error");
                grid[j, i].Selected = true;
            }
        }
    }
    foreach (var number in matrix)
    {
        int k = MultipleDigitCount(number);
        result[k]++;
    }
}

```

```

    }
    for (int i = 8; i >= 0; i--)
    {
        result[i] += result[i + 1];
    }
    return result;
}

private void task2Button_Click(object sender, EventArgs e)
{
    int[] a = GetNumbers(dataGridView1);
    int[] b = GetNumbers(dataGridView2);
    var form2 = new Form2(a, b);
    form2.Show();
}
}
}

```

Текст 2-го модуля:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;

namespace L2Sample
{
    public partial class Form2 : Form
    {
        private int[] array1;
        private int[] array2;
        public Form2(int[] array1, int[] array2)
        {
            this.array1 = array1;
            this.array2 = array2;
            InitializeComponent();
        }

        private void Form2_Load(object sender, EventArgs e)
        {
            if (array1.Length < array2.Length)
            {
                int[] tmp = array2;
                array2 = array1;
                array1 = tmp;
            }
            else
            {
                chart1.Series.Clear();
                var series1 = new Series() { ChartType =
SeriesChartType.Column, Color = Color.Red, LegendText = "Matrix 1"};

```

```

var series2 = new Series() { ChartType =
SeriesChartType.Column, Color = Color.Blue, LegendText = "Matrix 2" };
for (int i = 0; i < array2.Length; i++)
{
    series1.Points.AddXY(i, array1[i]);
    series2.Points.AddXY(i, array2[i]);
}
for (int i = array2.Length; i < array1.Length; i++)
{
    series1.Points.AddXY(i, array1[i]);
    series2.Points.AddXY(i, 0);
}
chart1.Series.Add(series1);
chart1.Series.Add(series2);
}
}
}
}
}

```

Вигляд вікон програми:

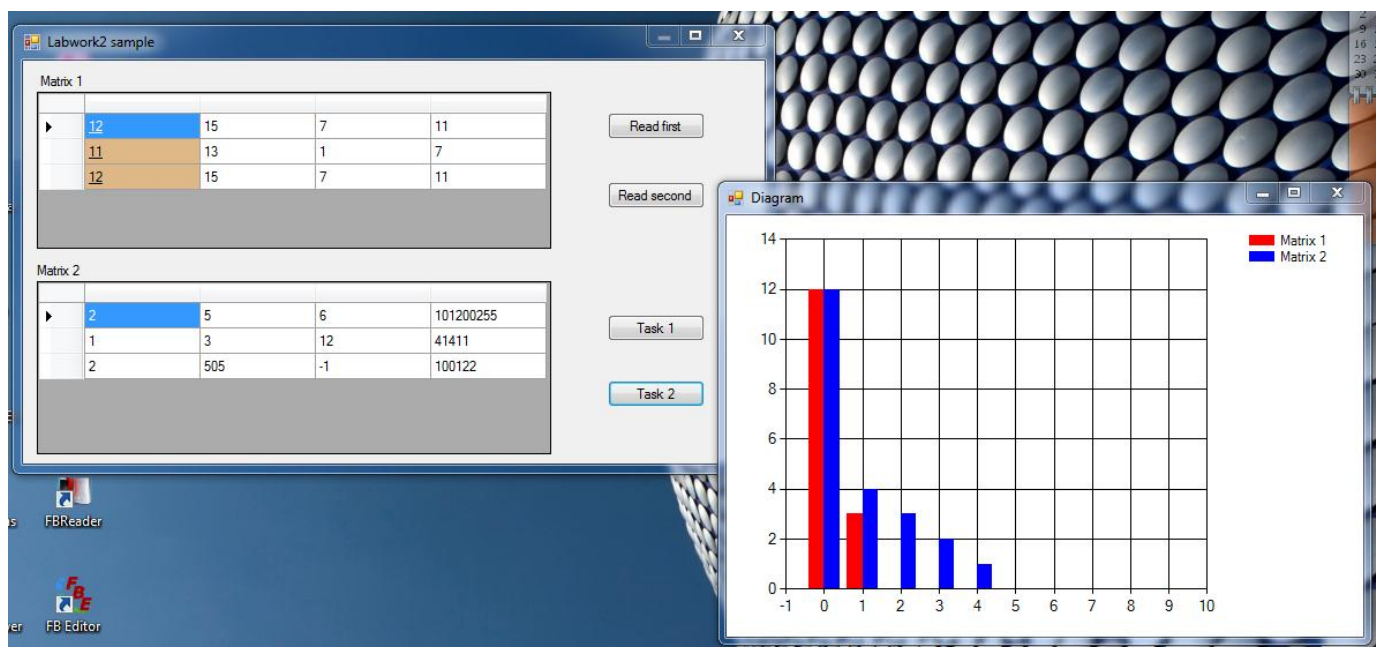


Рис. 2. Вигляд вікон програми

Варіанти завдань

1. Виділити курсивом елементи числової матриці, які можна записати у вигляді суми усіх тих елементів, які розташовані у сусідніх клітинках матриці. Зобразити діаграму, на осі абсцис якої відкладаються номери стовпців, а на осі ординат — середнє арифметичне додатних елементів цих стовпців.
2. Виділити кольором ті діагоналі числової матриці, які не містять жодного цілого елемента. Зобразити лінійну діаграму, поставивши кожному рядку у відповідність

окремий ряд даних. На осі абсцис треба відкласти номери стовпців, а на осі ординат — кількість додатних елементів цього стовпця, які розташовані у рядках від k -го і до останнього, де k — абсциса відповідної точки графіку.

3. Користувач вказує три файли, які містять елементи прямокутників числових матриць. Перевірити, чи є одна із матриць добутком двох інших. Зобразити лінійну діаграму, поставивши кожній матриці у відповідність окремий ряд даних. На осі абсцис треба відкласти натуральні числа від 1 до M , а на осі ординат — кількість елементів матриці, які не більші за k , де k — абсциса точки графіку.
4. Виділити кольором фон тих стовпців цілочислової матриці, усі елементи яких є попарно взаємно простими. Зобразити кругову діаграму, сектори якої відповідають рядкам матриці, а значення рівні відсотку, який складають додатні елементи серед усіх елементів рядка.
5. Підкреслити ті елементи числової матриці, які мають найбільшу дробову частину у своєму стовпці та рядку. Зобразити лінійну діаграму, поставивши кожному рядку у відповідність окремий ряд даних. На осі абсцис треба відкласти номери стовпців, а на осі ординат — суму елементів відповідного рядка, які розташовані у стовпцях від першого від 1-го до k -го, де k — абсциса відповідної точки графіку.
6. Користувач вказує імена N файлів, які містять елементи прямокутних числових матриць. Перевірити, чи є одна із матриць сумою усіх інших матриць. Зобразити лінійну діаграму, поставивши кожній матриці у відповідність окремий ряд даних. На осі абсцис треба відкласти натуральні числа від 2 до M , а на осі ординат — найбільший серед тих елементів відповідної матриці, сума індексів яких не перевищує k , де k — абсциса точки графіку.
7. Виділити напівжирним шрифтом елементи цілочислової матриці, які є простими числами. Зобразити діаграму, на осі абсцис якої відкладаються номери стовпців, а на осі ординат — кількість простих чисел у відповідному стовпці.
8. Рядки та стовпці числової матриці, які містять тільки невід’ємні елементи, виділити синім та жовтим кольором відповідно (перетин “синього” рядка та “жовтого” стовпця виділяти зеленим кольором). Зобразити діаграму, на осі абсцис треба відкласти номери стовпців, а на осі ординат — кількість нулів у цьому стовпці.
9. Користувач вказує чотири файли, які містять елементи прямокутних числових матриць. З’ясувати для кожної матриці, чи є вона сумою двох матриць з інших різних

- файлів, та повідомити назву файлу суми та файлів доданків. Зобразити лінійну діаграму, поставивши кожній матриці у відповідність окремий ряд даних. На осі абсцис треба відкласти натуральні числа від 1 до N , а на осі ординат — кількість нулів матриці, у стовпцях з номерами не більшими за k , де k — абсциса точки графіку.
10. Виділити кольором фон тих елементів цілочислової матриці, які є середнім арифметичним елементів свого стовпця). Зобразити кругову діаграму, сектори якої відповідають рядкам матриці, а значення рівні відсотку, який складають парні елементи серед усіх елементів рядка.
11. Підкреслити ті елементи числової матриці, які відрізняються від найближчого цілого числа не більше ніж на задане значення. Зобразити лінійну діаграму, поставивши кожному рядку у відповідність окремий ряд даних. На осі абсцис треба відкласти номери стовпців, а на осі ординат — кількість нулів відповідного рядка, які розташовані у стовпцях від першого від 1-го до k -го, де k — абсциса відповідної точки графіку.
12. Користувач вказує імена N файлів, які містять елементи квадратних цілочислових матриць порядку M . Вивести пари імен тих файлів, що містять матриці з однаковими сумами діагональних елементів. Зобразити лінійну діаграму, поставивши кожній матриці у відповідність окремий ряд даних. На осі абсцис треба відкласти цілі числа i від $1 - N$ до $N - 1$, а на осі ординат — суму елементів матриці, різниця номера рядка та номера стовпця яких рівна i .
13. Виділити зеленим кольором діагональні елементи числової прямокутної матриці N на M , синім — елементи над діагоналлю, жовтим — елементи під діагоналлю. Зобразити лінійну діаграму, поставивши елементам одного кольору у відповідність окремий ряд даних. Зобразити діаграму, на осі абсцис треба відкласти натуральні числа i від 2 до $N+M$, а на осі ординат — кількість додатніх елементів з сумою індексів не більшою за i .
14. Виділити ті рядки матриці, які не містять слів, утворених із латинських літер. Зобразити кругову діаграму, кожний елемент даних якої відповідає рядку таблиці, а значення — кількість різних слів у рядку.
15. Виділити курсивом елементи числової матриці, які можна записати у вигляді добутку сусідніх елементів матриці. Зобразити діаграму, на осі абсцис якої відкладаються номери рядків, а на осі ординат — кількість від'ємних елементів цих рядків.

16. Виділити кольором ті стовпці матриці, у яких нема жодного порожнього елемента. Зобразити лінійну діаграму, на осі абсцис якої треба відкласти номери стовпців, а на осі ординат — довжину найдовшого слова відповідного стовпця.
17. Користувач вказує три файли, які містять елементи прямокутників таблиць. Перевірити, чи є елементи однієї із таблиць конкатенацією елементів інших таблиць. Зобразити лінійну діаграму, поставивши кожній таблиці у відповідність окремий ряд даних. На осі абсцис треба відкласти натуральні числа k від 1 до M , а на осі ординат — кількість слів таблиці, довжина яких рівна k .
18. Виділити кольором ті стовпці матриці, усі елементи яких — дати. Зобразити гістограму, на осі абсцис якої треба відкласти номери стовпців, а на осі ординат — кількість комірок стовпця, які містять дані, які не є датами.
19. Виділити кольором монотонні рядки числової матриці (усі елементи розташовані або у порядку зростання, або у порядку спадання). Зобразити гістограму, на осі абсцис якої треба відкласти номери рядків, а на осі ординат — найбільшу серед цифр усіх чисел відповідного рядка.
20. Виділити кольором ті елементи матриці, які більші хоча би за половину своїх сусідів. Зобразити гістограму, на осі абсцис якої треба відкласти різні елементи матриці, а на осі ординат — кількості їх входжень.
21. Користувач вказує два файли, які містять елементи числових матриць. Виділити кольором ті елементи першої матриці, які більші за відповідні елементи другої матриці. Зобразити лінійну діаграму, поставивши кожній матриці у відповідність окремий ряд даних. На осі абсцис треба відкласти натуральні числа від 1 до M , а на осі ординат — суму усіх цифр елементів k -го рядка, де k — абсциса точки графіку, M — кількість рядків відповідної матриці.
22. Виділити кольором ті стовпці числової матриці, елементи яких розташовані у порядку спадання. Зобразити гістограму, на осі абсцис якої треба відкласти номери стовпців, а на осі ординат — суму цифр найменшого додатного елемента стовпця.
23. Виділити кольором ті стовпці числової матриці, сума елементів яких є степенем трійки (з цілим показником). Зобразити лінійний графік, позначивши на ньому точки, абсциса яких відповідає номеру стовпця, а ордината — середньому геометричному модулю елементів цього стовпця.

24. Виділити кольором ті рядки числової матриці, елементи яких утворюють геометричну прогресію. Зобразити графік із японських свічок, поставивши у відповідність кожному рядку свою свічку із відповідним максимальним, мінімальним, початковим та кінцевим значенням.
25. Користувач вказує два файли, які містять елементи матриць. Виділити кольором ті елементи першої матриці, які співпадають із відповідними елементами другої матриці. Зобразити лінійну діаграму, поставивши кожній матриці у відповідність окремий ряд даних. На осі абсцис треба відкласти натуральні числа від 1 до M , а на осі ординат — кількість елементів матриці, довжина яких рівна k , де k — абсциса точки графіку, M — довжина найдовшого елемента матриці.
26. Виділити курсивом унікальні елементи матриці (усі елементи, які жодного разу не повторюються). Зобразити гістограму, на осі абсцис якої треба відкласти номери рядків, а на осі ординат — суму унікальних елементів цього рядка.
27. Виділити кольором ті діагональні елементи числової матриці, які найбільші у своєму рядку та стовпчику. Зобразити на діаграмі точки, абсциси яких рівні номерам рядків матриці, а ординати — кількості різних елементів цих рядків.
28. Підкреслити вміст тих рядків числової матриці, елементи яких утворюють арифметичну прогресію. Зобразити гістограму, поставивши у відповідність кожному рядку прямокутник, висота якого рівна медіані модулів елементів рядка.
29. Виділити кольором ті стовпці числової матриці, максимальний елемент яких є степенем двійки (з цілим показником). Зобразити лінійний графік, позначивши на ньому точки, абсциса яких відповідає номеру стовпця, а ордината — середньому арифметичному додатних елементів цього стовпця.
30. Користувач вказує два файли, які містять елементи матриць. Виділити кольором ті рядки першої матриці, кожний елемент яких має більшу довжину ніж відповідний йому елемент другої матриці, розташований у тій самій позиції. Зобразити гістограму, поставивши кожній матриці у відповідність окремий ряд даних. На осі абсцис треба відкласти натуральні числа від 1 до M , а на осі ординат — кількість елементів матриці, до складу яких входить не менше, ніж k різних літер, де k — абсциса точки графіку, M — максимальна кількість різних літер у одному елементі матриці.

Лабораторна робота № 3

Теми:

1. Основи створення Web-сайтів за допомогою технології ASP.NET.
2. Використання колекцій.

Вимоги до роботи:

- у процесі написання коду обов'язково мають бути задіяні списки та асоціативні масиви;
- завдання треба виконати із використанням Web-форм;
- файли з даними потрібно зберігати на сервері та вивести їх вміст на сторінку;
- необхідно передбачити сторінку для авторизації користувачів;
- результати потрібно виводити на окремій сторінці;

Приклад. Написати Web-додаток для візуалізації інформації про лікарів, пацієнтів та прийоми. Створити на серверній стороні додатку код по знаходженню для кожного лікаря середнього віку його пацієнтів, які були у нього на прийомі у заданому користувачем році.

Клас DataProvider (модель даних):

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web;

namespace WebApplication10
{
    public static class DataProvider
    {
        public class Doctor
        {
            public int Id { get; set; }
            public string Name { get; set; }
            public string Speciality { get; set; }
        }

        public class Patient
        {
            public int Id { get; set; }
            public string Name { get; set; }
            public DateTime Birthday { get; set; }
        }

        public class Visit
        {
```

```

        public int PatientId { get; set; }
        public int DoctorId { get; set; }
        public string Date { get; set; }
    }
    private static List<Doctor> doctorList;
    private static List<Patient> patientList;
    private static List<Visit> visitList;

    public static List<Doctor> DoctorList
    {
        get
        {
            return doctorList;
        }
    }

    public static List<Patient> PatientList
    {
        get
        {
            return patientList;
        }
    }

    public static List<Visit> VisitList
    {
        get
        {
            return visitList;
        }
    }

    private const string doctorFileName = @"c:\doctors.txt";
    private const string patientFileName = @"c:\patients.txt";
    private const string visitFileName = @"c:\visits.txt";

    public static Dictionary<int, Doctor> DoctorDictionary;
    public static Dictionary<int, Patient> PatientDictionary;

    public static void ReadDate()
    {
        doctorList = new List<Doctor>();
        DoctorDictionary = new Dictionary<int, Doctor>();
        patientList = new List<Patient>();
        PatientDictionary = new Dictionary<int, Patient>();
        visitList = new List<Visit>();
        using(var reader = new StreamReader(doctorFileName))
        {
            while (!reader.EndOfStream)
            {
                var words = reader.ReadLine().Split(new[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);
                var doctor = new Doctor()
                {
                    Id = int.Parse(words[0]),

```

```

            Name = words[1],
            Speciality = words[2]
        };
        doctorList.Add(doctor);
        DoctorDictionary.Add(doctor.Id, doctor);
    }
}
using (var reader = new StreamReader(patientFileName))
{
    while (!reader.EndOfStream)
    {
        var words = reader.ReadLine().Split(new[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);
        var patient = new Patient()
        {
            Id = int.Parse(words[0]),
            Name = words[1],
            Birthday = DateTime.Parse(words[2])
        };
        patientList.Add(patient);
        PatientDictionary.Add(patient.Id, patient);
    }
}
using (var reader = new StreamReader(visitFileName))
{
    while (!reader.EndOfStream)
    {
        var words = reader.ReadLine().Split(new[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);
        visitList.Add(new Visit()
        {
            PatientId = int.Parse(words[0]),
            DoctorId = int.Parse(words[1]),
            Date = words[2]
        });
    }
}
}
}
}

```

Вміст файлу WebForm2.aspx.cs (основна програмна логіка):

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication10
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

```

```

if (!IsPostBack)
{
    DataProvider.ReadDate();
    DoctorsGridView.DataSource = DataProvider.DoctorList;
    DoctorsGridView.DataBind();
    PatientsGridView.DataSource = DataProvider.PatientList;
    PatientsGridView.DataBind();
    VisitsGridView.DataSource = DataProvider.VisitList;
    VisitsGridView.DataBind();
    var years = new SortedSet<int>();
    foreach (var visit in DataProvider.VisitList)
        years.Add(DateTime.Parse(visit.Date).Year);
    YearsDropDownList.Items.Clear();
    foreach (var year in years)
        YearsDropDownList.Items.Add(year.ToString());
}
}

class Pair
{
    public int TotalAge { get; set; }
    public int Count { get; set; }
}
protected void RunButton_Click(object sender, EventArgs e)
{
    ResultListBox.Items.Clear();
    int year = int.Parse(YearsDropDownList.SelectedValue);
    var doctorInfo = new Dictionary<int, Pair>();
    foreach (var visit in DataProvider.VisitList)
    {
        if (DateTime.Parse(visit.Date).Year == year)
        {
            if (doctorInfo.ContainsKey(visit.DoctorId))
            {
                doctorInfo[visit.DoctorId].TotalAge +=
                    DateTime.Today.Year -
DataProvider.PatientDictionary[visit.PatientId].Birthday.Year;
                doctorInfo[visit.DoctorId].Count++;
            }
            else
                doctorInfo[visit.DoctorId] = new Pair
                {
                    TotalAge = DateTime.Today.Year -
DataProvider.PatientDictionary[visit.PatientId].Birthday.Year,
                    Count = 1
                };
        }
    }
    foreach (var pair in doctorInfo)
    {
        ResultListBox.Items.Add(string.Format("{0}: {1}",
            DataProvider.DoctorDictionary[pair.Key].Name,
            (double)pair.Value.TotalAge / pair.Value.Count));
    }
}
}
}

```

Вигляд сторінки додатку:

| Id | Name | Speciality |
|----|----------|------------|
| 1 | Степанов | Терапевт |
| 2 | Іваненко | Хірург |
| 3 | Василюк | Кардіолог |

| Id | Name | Birthday |
|----|----------|--------------------|
| 1 | Петренко | 20.10.1998 0:00:00 |
| 2 | Іванчук | 10.05.1969 0:00:00 |
| 3 | Іванишин | 11.09.1975 0:00:00 |
| 4 | Лінтур | 04.04.1986 0:00:00 |

| PatientId | DoctorId | Date |
|-----------|----------|------------|
| 1 | 1 | 12.11.2016 |
| 2 | 2 | 30.09.2017 |
| 3 | 3 | 17.10.2015 |
| 2 | 1 | 30.09.2017 |
| 1 | 1 | 12.11.2016 |
| 2 | 1 | 30.09.2017 |
| 4 | 2 | 17.10.2015 |
| 4 | 1 | 30.09.2017 |
| 1 | 2 | 12.11.2016 |
| 2 | 1 | 30.09.2017 |
| 3 | 2 | 17.10.2015 |
| 2 | 1 | 30.09.2017 |
| 1 | 1 | 12.11.2016 |
| 3 | 3 | 30.09.2017 |
| 1 | 2 | 17.10.2015 |
| 1 | 3 | 30.09.2017 |

Year

Result
Василюк: 42
Іваненко: 30,66666

Рис. 3. Сторінка з результатом роботи програми

Варіанти завдань

- Файл «Студенти» містить наступну інформацію про студентів: код, прізвище, група. Файл «Рейтинг» містить коди студентів, назви предметів, кількість балів за перший модуль, кількість балів за другий модуль.
 - Вивести прізвища та назви груп студентів, які мають оцінку 4 із заданого предмету.
 - Вивести прізвища студентів, які мають найбільший середній рейтинг за перший модуль.
- Файл «Бібліотека» має наступну структуру: код книги, назва, автор. У файлі «Читачі» зберігаються прізвища читачів, коди виданих читачам книг, дати видачі, дати повернення. Розв'язати наступні задачі:
 - Вивести назви книг заданого автора, що були видані заданому читачу протягом вказаного діапазону дат.

- b) Вивести назви книг, відсоток неповернутих примірників (відношення кількості неповернутих книг до загальної кількості примірників цієї книги) яких найбільший.
3. Файл «Лікарі» має наступну структуру: прізвище, спеціалізація. Рядки файлу «Пацієнти» містять прізвище пацієнта, діагноз, прізвище лікаря, відділення, дата надходження, дата виписки.
- a) Вивести кількість різних діагнозів, які поставили лікарі із заданою спеціалізацією.
- b) Вивести прізвища лікарів із найбільшою кількістю невиписаних пацієнтів у заданому відділенні.
4. Файл «Товари» має наступну структуру: код товару, назва товару (коди унікальні, назви можуть повторюватися). Рядки файлу «Поставки» містять код товару, дату, назву постачальника, обсяг поставок, ціну за одиницю.
- a) Вивести середній обсяг поставок товару із заданим кодом, здійснених протягом другого півріччя минулого року, ціна товару в яких не менша за задане значення.
- b) Вивести постачальників, які виконали найменшу кількість поставок товару із заданою назвою.
5. Файл «Жильці» містить інформацію у такому форматі: код жильця, прізвище жильця, адреса. Кожний рядок файлу «Оплата» містять інформацію про сплату за послуги у такій формі: код жильця, вид послуги, (вода, газ, ...), сплачена сума, дата сплати. Розв'язати наступні задачі:
- a) Знайти середні сплати за газ у поточному місяці усіх жильців, які проживають на заданій вулиці.
- b) Вивести назву послуги, за яку загалом сплачено найбільшу суму за останній квартал.
6. Файл «Учні» має наступну структуру: код учня, прізвище учня, школа. Кожний рядок файлу «ЗНО» містить код учня, назву предмета, дату складання та бал. Розв'язати наступні задачі:
- a) Для кожного школяра заданої школи вивести назви предметів, з яких він складав ЗНО.
- b) Вивести прізвища учнів з трьома найбільшими середніми балами у заданому році.
7. Файл «Готель» має наступну структуру: номер кімнати, поверх, кількість місць у кімнаті, вартість проживання за одну добу. У файлі «Жильці» містяться прізвища жильців, номери кімнати, дати поселення та виселення. Розв'язати наступні задачі:

- a) Вивести поверхи з найменшою сумарною кількістю місць.
 - b) Для кожного жильця, який ще не виписався з готелю, вивести грошову суму, яку він має сплатити за проживання (по сьогоднішній день).
8. Файл «Авіарейси» має наступну структуру: номер рейсу, пункт відправлення, час вильоту, загальна кількість місць. У рядках файлу «Квитки» міститься номер квитка, номер рейсу, місце, дата вильоту, пункт призначення, дата та час прибуття, ціна. Розв'язати наступні задачі:
- a) Вивести пункти призначення, у які не продано жодного квитка на рейси, які відлітають з Києва після обіду.
 - b) Вивести кількість вільних місць по усім рейсам (із урахуванням кількості проданих квитків).
9. Файл «Завод» має наступну структуру: код працівника, прізвище, стать, номер цеху, посада, стаж, оклад. Файл «Премія» має наступну структуру: код, премія. Розв'язати наступні задачі:
- a) Вивести кількість посад працівниць, які мають премію.
 - b) Для кожного цеху вивести максимальну зарплату (оклад + сумарна премія) його працівників, стаж яких потрапляє у межі від 10 до 20.
10. Файл «Студенти» має наступну структуру: код студента, прізвище, група, курс. Файл «Участь у гуртках» містить код студента та назву гуртка.
- a) Вивести чотирьох студентів, які відвідують найбільшу кількість гуртків.
 - b) Вивести кількість студентів 1-го та 3-го курсів, які займаються у гуртку із заданою назвою.
11. Файл «Автоперевезення» містить наступну інформацію: номер автобази, номер машини, дата перевезення, назва вантажу, вартість перевезення.
- a) Вивести назви вантажів, які жодного разу не перевозилися автомобілем із заданим номером.
 - b) Вивести номер автобази, сумарна вартість перевезень, виконаних автомобілями цієї бази, є максимальною у заданому діапазоні дат.
12. Файл «Студенти» містить наступну інформацію про студентів: код, прізвище, група. Файл «Рейтинг» містить коди студентів, назви предметів, форму контролю (залік або іспит), підсумковий бал.
- a) Вивести прізвища та назви груп студентів, які мають хоча би одну двійку.

- b) Вивести прізвища студентів та їх підсумковий рейтинг, впорядкувавши запису по рейтингу у порядку спадання.
13. Файл «Бібліотека» має наступну структуру: код книги, назва, автор, рік видання. У файлі «Читачі» зберігаються прізвища читачів, коди виданих читачам книг, дати видачі, дати повернення. Розв'язати наступні задачі:
- a) Вивести прізвища читачів, які вибрали книги минулого року та ще їх не повернули.
- b) Вивести коди та назви книг і загальну кількість видач та повернень цих книг.
14. Файл «Лікарі» має наступну структуру: прізвище лікаря, спеціалізація. Рядки файлу «Лікарня» містять прізвище пацієнта, прізвище лікаря, діагноз, дата надходження, дата виписки.
- a) Вивести кількість спеціалізацій лікарів, які ставили діагнози заданому пацієнту у вказаному році.
- b) Вивести список прізвищ пацієнтів та сумарну тривалість їх перебування у лікарні.
15. Файл «Товари» має наступну структуру: код товару, назва товару (коди унікальні, назви можуть повторюватися), ціна одиниці товару. Рядки файлу «Поставки» містять код товару, дату поставки, обсяг поставки (кількість одиниць товару у поставці).
- a) Вивести максимальний обсяг поставок товару із заданою назвою.
- b) Вивести середні загальні вартості поставок товарів по кожному року (загальна вартість — це добуток ціни одиниці товару на обсяг поставки).
16. Файл «Жильці» містить інформацію у такому форматі: код жильця, прізвище жильця, адреса. Кожний рядок файлу «Оплата» містять інформацію про сплату за послуги у такій формі: код жильця, вид послуги, (вода, газ, ...), сплачена сума, дата сплати. Розв'язати наступні задачі:
- a) Знайти сумарні сплати за газ у заданому році жильцями, які проживають за заданою адресою.
- b) Вивести прізвища жильців, які загалом сплатили найбільшу суму за останній рік.
17. Файл «Учні» має наступну структуру: код учня, прізвище учня, школа. Файл «ЗНО» містить коди учнів, назви предметів, дати складання та бали. Розв'язати наступні задачі:
- a) Для кожного школяра вивести кількість предметів, з яких він склав ЗНО не менше, ніж на 180 балів.
- b) Вивести назви шкіл та середній бал їх учнів у заданому році.

18. Файл «Готель» має наступну структуру: номер кімнати, поверх, кількість місць у кімнаті, вартість проживання за одну добу. У файлі «Жильці» містяться прізвища жильців, номери кімнати, дати поселення та виселення. Розв'язати наступні задачі:
- Вивести прізвища жильців, які проживали заданого числа на заданому поверсі.
 - Для кожного поверху вивести максимальну грошову суму, яку має сплатити за добу вже поселений, але ще не виселений мешканець цього поверху.
19. Файл «Авіарейси» має наступну структуру: номер рейсу, пункт відправлення, пункт прибуття, час відправлення, час прибуття, ціна. У рядках файлу «Квитки» міститься номер квитка, номер рейсу, дата, місце. Розв'язати наступні задачі:
- Вивести пункти призначення, у які придбано квитки у заданий день на задане місце.
 - Вивести номери рейсів, їх тривалість та загальну кількість проданих на рейс квитків протягом останнього року, впорядкувавши записи по тривалості рейсів.
20. Файл «Завод» має наступну структуру: код працівника, прізвище, стать, посада, оклад. Файл «Премія» має наступну структуру: код, дата, премія. Розв'язати наступні задачі:
- Вивести назви посад, працівники яких у поточному році жодного разу не отримали премії.
 - Для кожної посади вивести середню зарплату (оклад + сумарна премія) його працівників-чоловіків.
21. Файл «Студенти» має наступну структуру: код студента, прізвище, курс. Файл «Участь у гуртках» містить код студента, назву гуртка та рік проведення.
- Вивести кількість студентів заданого курсу, які займаються у гуртку із заданою назвою.
 - Вивести роки, у які студент із заданим прізвищем брав участь у найбільшій кількості гуртків.
22. Файл «Автоперевезення» містить наступну інформацію: номер машини, дата перевезення, назва вантажу, вартість перевезення. Файл "Машини" містить номер машини та прізвище водія.
- Вивести назви вантажів, які частіше за все перевозив водій із заданим прізвищем.
 - Вивести прізвища водіїв, сумарна вартість перевезень яких у поточному році є найбільшою.

23. Файл «Студенти» містить наступну інформацію про студентів: код, дата народження, прізвище, група. Файл «Рейтинг» містить коди студентів, назви предметів, кількість балів за перший модуль, кількість балів за другий модуль.
- Вивести без повторів назви груп, студенти яких не мають жодної двійки із заданого предмета.
 - Вивести прізвища студентів заданого року народження, середній бал яких за перший модуль рівний середньому балу за другий модуль.
24. Файл «Книги» має наступну структуру: код книги, назва, автор, рік видання. У файлі «Читачі» зберігаються прізвища читачів, коди виданих читачам книг, дати видачі, дати повернення. Розв'язати наступні задачі:
- Вивести прізвища читачів, які принаймні один раз у позаминулому році брали книги заданого автора.
 - Вивести назви книг, кількість неповернутих примірників яких найбільша.
25. Файл «Лікарі» має наступну структуру: прізвище, відділення. Рядки файлу «Пацієнти» містять прізвище пацієнта, діагноз, прізвище лікаря, дата прийому.
- Вивести прізвища пацієнтів, яких приймали лікарі заданого відділення у вказаному році.
 - Вивести назви діагнозів та кількості різних відділень, лікарі яких поставили цей діагноз, впорядкувавши записи по назві діагнозу.
26. Файл «Товари» має наступну структуру: код товару, назва товару, ціна одиниці товару (коди унікальні, назви можуть повторюватися). Рядки файлу «Поставки» містять код товару, дату, назву постачальника, обсяг поставок (кількість одиниць).
- Вивести сумарну вартість усіх поставок товару із заданою назвою, здійснених у заданому діапазоні дат.
 - Вивести назви постачальників та кількість різних товарів, які поставив цей постачальник, впорядкувавши записи по кількості товарів.
27. Файл «Жильці» містить інформацію у такому форматі: код жильця, прізвище жильця, адреса. Кожний рядок файлу «Оплата» містять інформацію про сплату за послуги у такій формі: код жильця, вид послуги, (вода, газ, ...), сплачена сума, дата сплати. Розв'язати наступні задачі:
- Знайти найменші сплати за надання заданої послуги жильцями, які проживають у вказаному місті.

б) Для кожного жителя вивести його сумарні сплати за усі послуги за кожний рік.

28. Файл «Учні» має наступну структуру: код учня, прізвище учня, школа. Кожний рядок файлу «ЗНО» містить код учня, назву предмета, дату складання та бал. Розв'язати наступні задачі:

а) Вивести прізвища учнів, їх школи, предмет та бал, включивши у результат дані про складання ЗНО у заданому році.

б) Для кожного року, дані по якому присутні у файлах, вивести кількість учнів заданої школи, які склали ЗНО із заданого предмету.

29. Файл «Готель» має наступну структуру: номер кімнати, поверх, кількість місць у кімнаті, вартість проживання за одну добу. У файлі «Жильці» містяться прізвища жильців, номери кімнати, дати поселення та виселення. Розв'язати наступні задачі:

а) Вивести прізвища, вартість, поверх та номер кімнати тих жильців, які ще не виселилися.

б) Вивести номери кімнат та кількість вільних місць у них.

30. Файл «Завод» має наступну структуру: код працівника, прізвище, стать, номер цеху, посада, оклад. Файл «Премія» має наступну структуру: код, премія, дата. Розв'язати наступні задачі:

а) Вивести номери цехів, жоден працівник-чоловік у яких не має премію у поточному році.

б) Для кожної посади вивести сумарну зарплату (оклад + сумарна премія) його працівників.