

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет “Львівська політехніка”



РОБОТА З МАСИВАМИ В С

ІНСТРУКЦІЯ

до лабораторної роботи № 3 з курсу
“Основи програмування”
для базового напрямку “Програмна інженерія”

Затверджено
На засіданні кафедри
програмного забезпечення
Протокол № від

ЛЬВІВ – 2018

1. МЕТА РОБОТИ

Мета роботи – навчитися організовувати такі структури даних як масиви та освоїти основні методи програмування алгоритмів обробки масивів даних засобами мови C.

2. ТЕОРЕТИЧНІ ВІДОМОСТІ

Масив являє собою сукупність однотипних змінних, розміщених у послідовно пронумерованих суміжних комірках пам'яті. Номер елемента масиву задається індексом. **Індексація елементів масиву в C починається з 0.** Якщо у масиві N елементів, то перший елемент матиме індекс 0, а останній – індекс (N-1).

За способом зв'язування індексів з комірками пам'яті виділяють три категорії масивів: *статичні, фіксовані автоматичні та динамічні.*

Статичним називають масив, в якому зв'язування індексів та розміщення в пам'яті виконуються на етапі компіляції програми. Статичні масиви досить ефективні, оскільки для їх створення та знищення не потрібно додаткових операцій.

Фіксованим автоматичним називають масив, в якому індекси зв'язуються статично, а розміщення в пам'яті виконується при обробці оголошень всередині функцій.

Динамічним називають масив, де зв'язування індексів і розміщення в пам'яті виконується безпосередньо під час виконання програми. Цей спосіб організації даних достатньо гнучкий, оскільки розміри динамічного масиву можуть збільшуватися та зменшуватися під час виконання програми у міру необхідності.

Об'єм пам'яті, який займає масив, рівний добутку розміру типу елементів масиву та кількості елементів в масиві. Наприклад, якщо масив містить 16 чисел типу `int` і тип `int` займає в пам'яті 2 байти, то цілий масив займатиме $2 \times 16 = 32$ байти.

У мові C не перевіряється вихід індексу масиву за межі допустимого значення. Вихід за межі пам'яті, яку займає масив, не контролюється. При помилці програміста може відбутися запис інформації в сусідні комірки пам'яті, а **наслідки цього непередбачувані.**

Масиви можуть бути одновимірними та багатовимірними.

Одновимірні масиви

Одновимірні масиви оголошуються таким чином:

```
тип ім'я_масиву[розмір]
```

Наприклад, так оголошується статичний масив з іменем `numbs`, що складається з 10 елементів типу `int`:

```
int numbs[10];
```

У C доступ до елементів масиву здійснюється за допомогою спеціального оператора `[]`. В квадратних дужках вказується індекс елемента масиву, а перед ними – його ім'я. Наприклад, присвоєння першому елементу масиву `numbs` значення 12 виконується так:

```
numbs[0] = 12;
```

Масиви тісно пов'язані із вказівниками. Ім'я масиву є вказівником на його перший елемент. Тому ім'я масиву можна використати в якості бази для зміщення вказівника. Наприклад, вираз

```
numbs[4] = 1
```

еквівалентний виразу

```
*(numbs+4) = 1;
```

Ініціалізація масивів

Для заповнення масиву початковими даними достатньо виконати ініціалізацію під час оголошення масиву:

```
int nums[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```

При цьому не потрібно вказувати розмір масиву – він буде розпізнаний за кількістю введених елементів. Тому можна писати так:

```
int nums[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
```

Більше того, якщо вказати в квадратних дужках неправильний розмір, все одно компілятор автоматично обчислить правильний розмір. Наприклад, при оголошенні:

```
int nums[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8}
```

компілятор «вважатиме», що масив містить 9 елементів. Однак, при цьому виникне проблема при наступному присвоєнні:

```
nums[10] = 10;
```

Компілятор розпізнає помилку, коли присвоюється більше чисел, ніж задано елементів масиву:

```
int nums[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

При цьому видається повідомлення «Занадто багато ініціалізаторів».

До масивів застосовний оператор **sizeof**, який повертає розмір масиву у байтах:

```
sizeof(nums)
```

Якщо масив не підлягає змінам, його можна оголосити константним, додавши слово **const** в оголошення масиву:

```
const int nums[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

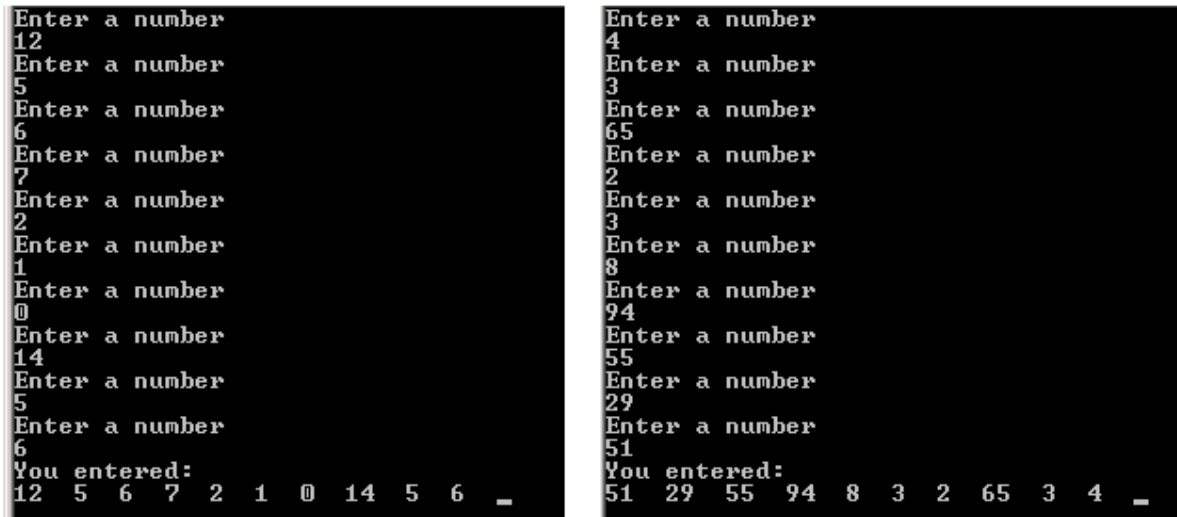
Приклад. 3 клавіатури ввести 10 цілих чисел, записавши їх у масив. Роздрукувати введені числа в рядок через два пробіли.

```
#include<stdio.h>
#include <conio.h>
int main(){
    int vect[10]; // оголошення масиву з 10 цілих чисел
    for(int i = 0; i < 10; i++)
        /* у циклі виводимо підказку про введення чисел */
        printf("Enter a number");
        scanf("%d", &vect[i]); /* на i-ій ітерації циклу записуємо введене число в
i-ий елемент масиву; змінна циклу має початкове значення 0, оскільки елементи масиву
індексуються починаючи з 0 */
    }
    printf("You entered:");
    for(int i = 0; i < 10; i++)
        /* у циклі виводимо всі введені числа, відокремлені двома пробілами */
        printf("%d ", vect[i]);
    }
    return 0;
    getch();
}
```

Приклад. З клавіатури ввести 10 цілих чисел, записавши їх у масив. Роздрукувати введені числа у зворотному порядку.

```
int vect[10];
for(int i = 0; i < 10; i++)
{
    printf("Enter a number");
    scanf("%d", &vect[i]);
}
printf("You entered:");
for(int i = 9; i >= 0; i--) /* елементи масиву «перебираємо» з кінця масиву */
{
    printf("%d ", vect[i]);
}
```

Результати показані на рис. 1



а)

б)

Рис. 1 Введення 10 цілих чисел та роздрук їх в одному рядку в прямому (а) та зворотному (б) порядку

Приклад. Ввести з клавіатури 10 символів. Порахувати кількості введених голосних та вивести їх на зразок таблиці.

```
const char vowels[] = {'a','e','i','o','u','y'}; /* оголошуємо масив vowels з 6
елементів типу char та одразу здійснюємо ініціалізацію латинськими голосними літерами;
масив оголошено константним тому, що не передбачається його змінювати протягом
виконання програми */
char usersymbols[10]; /* оголошуємо масив usersymbols, в який будемо записувати
10 символів, введених користувачем з клавіатури */
int frequencies[6]; /* оголошуємо масив frequencies з 6 цілих чисел, який
зберігатиме інформацію про те, скільки разів користувач ввів кожну з голосних літер;
тому кількість елементів масиву frequencies збігається з кількістю елементів масиву
vowels */
for(int i = 0; i < sizeof(usersymbols)/sizeof(char); i++)
/* в циклі заповнюємо масив usersymbols; вираз sizeof(usersymbols)/sizeof(char)
рівний числу елементів в масиві usersymbols */
    printf("Enter a symbol");
    scanf("%c",&usersymbols[i]);
}
for(int i = 0; i < sizeof(vowels)/sizeof(char); i++)
/* у циклі «перебираємо» всі голосні літери, спочатку вважаючи, що кожна з них
зустрілася серед введених користувачем символів 0 разів; i-ий елемент масиву vowels
відповідає i-ому елементу масиву frequencies */
    frequencies[i] = 0;
    for(int j = 0; j < sizeof(usersymbols)/sizeof(char); j++)
```

```

        { /* у внутрішньому циклі перевіряємо відповідність кожного введеного
символу i-ій голосній літері; якщо відповідність знайдена, збільшуємо i-ий елемент
масиву frequencies на 1*/
            if(usersymbols[j]==vowels[i])
                frequencies[i]++;
        }
    }
    for(int i = 0; i < sizeof(vowels)/sizeof(char); i++)
    { /* виводимо результати на екран у два стовпці */
        printf("%c %d", vowels[i], frequencies[i]);
    }
}

```

Результат показаний на рис. 2

```

Enter a symbol
w
Enter a symbol
e
Enter a symbol
a
Enter a symbol
y
Enter a symbol
o
Enter a symbol
e
Enter a symbol
a
Enter a symbol
o
Enter a symbol
o
Enter a symbol
l
a      2
e      2
i      0
o      3
u      0
y      1
_

```

Рис. 2 Підрахунок кількості введених з клавіатури голосних літер

Двовимірні масиви

Часто за допомогою мов програмування виникає потреба обробляти таблиці або матриці. Для їх представлення у мові С передбачені багатовимірні масиви.

Найпоширеніший варіант багатовимірного масиву – двовимірний масив, який можна представити у вигляді масиву одновимірних масивів:

тип ім'я_масиву [розмір1][розмір2]

Звертаються до елементів масиву так:

ім'я_масиву [індекс1][індекс2]

Наприклад:

```

int matr[10][5]; /* оголошується матриця з 10 рядків і 5 стовпців */
matr[4][3] = 10; /* четвертому елементу n'ятого рядка присвоюється значення 10 */

```

Ініціалізація двовимірного масиву здійснюється по рядках:

```

int nums[3][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9};

```

В результаті перший рядок міститиме числа 1, 2, 3, другий – числа 4, 5, 6, а третій – числа 7, 8, 9.

Для наочності краще при ініціалізації використовувати групування:

```
int nums[3][3] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
```

Двовимірний масив, як і будь-який інший, зберігається в послідовних комірках пам'яті, а тому його можна представити як одномірний, пам'ятаючи, що ім'я масиву – це вказівник на його перший елемент. Нехай двовимірний масив `mas` має розмірність $n*m$. Тоді елемент `mas[i][j]` можна представити як `*(mas+n*i+j)`. З іншого боку, вираз `mas[i]` є вказівником на перший елемент i -ого рядка, а тому `*[mas[i]+j]` – це j -ий елемент i -ого рядка.

Приклад. З клавіатури ввести квадратну матрицю. Роздрукувати її.

```
int matr[N][N];
for(int i = 0; i < N; i++)
    for(int j = 0; j < N; j++)
    {
        printf("Enter the %d element of %d row", (j+1), (i+1));
        scanf("%d", &matr[i][j]);
    }
for(int i = 0; i < N; i++)
    for(int j = 0; j < N; j++)
        printf("%d\t", matr[i][j]);
```

Розмір масиву має бути константою. Оголошення

```
int N = 4;
int matr[N][N];
```

привело б до помилки компіляції.

Хоча у C індексація елементів масивів починається з нуля, тим не менш зручно іменувати елемент з індексом 0 першим. Тому підказку для вводу масиву виводимо зі збільшеними на 1 індексами:

```
printf("Enter the %d element of %d row", (j+1), (i+1));
```

Введену матрицю виводимо по рядках, розмежовуючи елементи символами табуляції для покращення читабельності. Фрагмент результату показаний на рис. 3.

```
Enter the 3 element of 2 row
66
Enter the 4 element of 2 row
77
Enter the 1 element of 3 row
0
Enter the 2 element of 3 row
1
Enter the 3 element of 3 row
2
Enter the 4 element of 3 row
3
Enter the 1 element of 4 row
4
Enter the 2 element of 4 row
5
Enter the 3 element of 4 row
6
Enter the 4 element of 4 row
77
4      5      6      23
4      5      66     77
0      1      2      3
4      5      6      77
```

Рис. 3 Ввід та роздрук квадратної матриці

Оскільки масиви тісно пов'язані зі вказівниками, то програму для вводу та роздруку квадратної матриці цілих чисел можна переписати наступним чином:

```
int matr[N][N];
for(int i = 0; i < N*N; i++)
```

```

    {
        /* квадратна матриця містить N*N елементів, *matr є вказівником на перший
        елемент першого рядка матриці matr, а змінна i задає зміщення елемента від початку
        масиву, оскільки всі елементи розміщені в пам'яті послідовно */
        printf("Enter a number");
        scanf("%d", *matr+i);
    }

for(int i = 0; i < N; i++)
{
    for(int j = 0; j < N; j++)
        printf("%d\t", matr[i][j]);
}

```

Вивід елементів масиву matr в рядок можна здійснити так:

```

for(int i = 0; i < N*N; i++)
{
    printf ("%d\t", *matr+i);
}

```

Вивід у табличному вигляді можна здійснити лише з використанням одного циклу:

```

for(int i = 0; i < N*N; i++)
{
    printf("%d\t", *(matr+i));
    if(!((i+1)%N)) printf("\n");
}

```

За допомогою оператора if після кожних N виведених на екран елементів переходимо на новий рядок.

Приклад. З клавіатури ввести квадратну матрицю. Роздрукувати саму матрицю та номер рядка, сума елементів якого є максимальною (рядки для зручності нумеруватимемо з 1).

```

int matr[N][N];
for(int i = 0; i < N*N; i++)
{
    printf("Enter a number");
    scanf("%d", *matr+i);
}

int max = 0;
int maxn = 1;
int sum;
for(int i = 0; i < N; i++)
{
    sum = 0;
    for(int j = 0; j < N; j++)
        sum = sum + matr[i][j];
    if(sum > max) {max = sum; maxn = i;}
}
for(int i = 0; i < N*N; i++)
{
    printf("%d\t", *(matr+i));
    if(!((i+1)%N)) printf("\n");
}
printf("The maximum sum is in the row # %d ", (maxn+1));

```

Змінна max міститиме поточну значення максимальної суми елементів в рядку (спочатку присвоїмо max = 0). У змінній maxn зберігатимемо номер рядка, сума елементів якого є максимальною. У змінній sum зберігатимемо суму елементів рядка, який опрацьовується у даний момент. Перед опрацюванням кожного рядка змінну sum «скидаємо в нуль». Якщо змінна sum більша за поточне значення max, то max присвоюємо значення sum, а номер відповідного рядка записуємо в maxn. Результати показані на рис. 4.

```

Enter a number
1
Enter a number
2
Enter a number
3
Enter a number
2
Enter a number
4
Enter a number
8
Enter a number
1
Enter a number
1
Enter a number
0
Enter a number
4
Enter a number
9
Enter a number
9
Enter a number
1
Enter a number
2
Enter a number
1
Enter a number
0
1      2      3      2
4      8      1      1
0      4      9      9
1      2      1      0
The maximum sum is in the row # 3

```

Рис. 4 Роздрук квадратної матриці та визначення номера рядка з найбільшою сумою елементів

Приклад. Ввести з клавіатури квадратну матрицю. Роздрукувати її разом з сумою елементів кожного рядка.

```

int matr[N][N];
for(int i = 0; i < N*N; i++)
{
    printf("Enter a number");
    scanf("%d", *matr+i);
}
int max = 0;
int maxn = 1;
int sum;
int vect[N]; /* y vect[i] запишемо суму елементів i-ого рядка матриці matr*/
for(int i = 0; i < N; i++)
{
    sum = 0;
    for(int j = 0; j < N; j++)
        sum = sum + matr[i][j];
    if(sum > max) {max = sum; maxn = i;}
    vect[i] = sum;
}
for(int i = 0; i < N; i++)
{
    for(int j = 0; j < N; j++)
        printf("%d\t", matr[i][j]);
    printf("Suma: %d\n", vect[i]);
}

```

Результат показаний на рис. 5


```

Enter a number
1
Enter a number
2
Enter a number
3
Enter a number
2
Enter a number
1
Enter a number
2
Enter a number
6
Enter a number
2
Enter a number
1
Enter a number
1
Enter a number
1
Enter a number
2
Enter a number
3
Enter a number
2
Enter a number
9
Enter a number
9
1      2      3      2      Suma: 8
1      2      6      2      Suma: 11
1      1      1      2      Suma: 5
3      2      9      9      Suma: 23
The maximum sum is in the row # 4

```

Рис. 5 Ввід та роздрук квадратної матриці з сумою елементів кожного рядка

3. КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке масив? Які типи масивів Ви знаєте?
2. Який загальний вигляд опису одновимірного масиву? А двовимірного?
3. Що таке вказівник? Яка відмінність вказівника від звичайної змінної?
4. Яку роль відіграє ім'я масиву? Як можна його використати?
5. Як здійснюється ініціалізація масиву? Які особливості ініціалізації багатовимірних масивів?
6. Опишіть відомі Вам способи доступу до елементів одновимірного масиву.
7. Як можна організувати контроль виходу індексів за межі допустимого діапазону?
8. Як зберігаються в пам'яті багатовимірні масиви?
9. Як визначити скільки оперативної пам'яті займає масив?
10. Як здійснюється доступ до елементів багатовимірного масиву за допомогою індексів? А за допомогою вказівників?
11. Напишіть два різних оператора на C, які б присвоювали вказівнику *ptr* адресу першого елемента четвертого рядка двовимірного масиву *x[5][7]*.
12. Напишіть два різних оператора на C, які інкрементують шостий елемент масиву *array*, використовуючи механізм індексації та вказівника зі зміщенням, відповідно.

4. ЛАБОРАТОРНЕ ЗАВДАННЯ

1. Ознайомитися з теоретичним матеріалом викладеним вище в даній інструкції і виконати приклади програм.
2. Одержати індивідуальне завдання з Додатку 1.

3. Розробити алгоритм розв'язання індивідуального завдання і подати його у вигляді блок-схеми.
4. Скласти програму на мові С у відповідності з розробленим алгоритмом.
5. Виконати обчислення по програмі.
6. Одержати індивідуальне завдання з Додатку 2.
7. Розробити алгоритм розв'язання індивідуального завдання і подати його у вигляді блок-схеми.
8. Скласти програму на мові С у відповідності з розробленим алгоритмом.
9. Виконати обчислення по програмі при різних значеннях точності і порівняти отримані результати.
10. Підготувати та здати звіт про виконання лабораторної роботи.

5. СПИСОК ЛІТЕРАТУРИ

1. Керниган Б., Ритчи Д. Язык программирования С. - М. - Финансы и статистика. - 1992. - 272 с.
2. Уэйт М., Прата С., Мартин Д. Язык С. Руководство для начинающих. - М. - Мир. - 1988. - 512 с.
3. К. Джемса. Учимся программировать на языке С. М.: Мир, 1997. - 320 с
4. Герберт Шилдт. Полный справочник по С++. М. - С.-П.-К., Вильямс. - 2003. - 800 с.
5. Демидович Е. М. Основы алгоритмизации и программирования. Язык Си. (Учебное пособие). - Санкт-Петербург: "БХВ Петербург". - 2006. - 439 с.

6. ІНДИВІДУАЛЬНІ ЗАВДАННЯ

ДОДАТОК 1

Написати програму для обробки даних, організованих у масив, згідно завдання наведеного варіанту. У програмі використати такі форми звертання до елементів масиву:

- на основі індексів,
- через розадресоване ім'я масиву і зміщення елемента.

1. З клавіатури ввести вектор дійсних чисел. Визначити значення і порядкові номери двох найбільших елементів введеного вектора.
2. З клавіатури ввести два вектори цілих чисел. Утворити новий вектор із спільних елементів двох введених. У новому векторі не має бути повторень елементів. Надрукувати сформований вектор або вивести повідомлення про відсутність спільних елементів.
3. Перевірити, чи введений з клавіатури вектор дійсних чисел є спадним. Якщо ні – то визначити найменший елемент введеного вектора і поміняти його місцями з останнім.
4. Ввести вектор дійсних чисел. Всі елементи вектора, які розташовані за мінімальним, зменшити в 2,5 рази.
5. Задано масив цілих чисел. Циклічно зсунути його елементи на 2 позиції вправо. *Підказка:* циклічний зсув передбачає, що старші елементи масиву переходять на місце зсунутих молодших.
6. З клавіатури ввести масив дійсних чисел. Утворити новий масив, кожен елемент якого дорівнює півсумі значень двох сусідніх елементів введеного вектора. Визначити номер мінімального елемента нового вектора.
7. Ввести вектор дійсних чисел. Всі елементи вектора, що менші за його середнє арифметичне значення, збільшити в 2 рази.
8. Визначити адресу найбільшого числа та найменшого числа у введеному векторі дійсних чисел. Порахувати кількість нулів в заданому векторі.
9. Визначити суму парних елементів та добуток непарних елементів введеного з клавіатури вектора дійсних чисел.

10. Ввести вектор цілих чисел. Видрукувати адреси елементів введеного вектора, які при діленні на 10 мають остачу 3.
11. З клавіатури ввести вектор цілих чисел. Визначити значення двох найменших елементів вектора. Якщо вони не є однаковими, то замінити ці елементи значенням їх півсуми.
12. З клавіатури ввести вектор дійсних чисел. Знайти середнє арифметичне усіх чисел, що розміщені між максимальним та мінімальним елементами вектора.
13. З клавіатури ввести два вектори цілих чисел. Утворити новий вектор із спільних парних чисел двох введених та неспільних непарних чисел.
14. Ввести вектор дійсних чисел. Всі елементи вектора, що розміщені за його максимальним елементом, збільшити на середнє арифметичне елементів вектора.
15. Ввести два натуральних числа і десяткову цифру. Сформувати вектор з усіх натуральних чисел (але не більше, ніж 100), що потрапляють у проміжок між введеними числами і містять задану цифру.
16. Ввести вектор дійсних чисел. Всі від'ємні елементи збільшити на величину середнього арифметичного елементів вектора, а додатні зменшити на цю величину. Мінімальний та максимальний елемент вектора не змінювати.
17. Ввести вектор дійсних чисел. Всі елементи менші за величину середнього арифметичного елементів вектора помножити на 1,7, а нульові елементи замінити на значення середнього арифметичного.
18. Ввести вектор цілих чисел. Видрукувати адреси елементів вектора, які кратні 7.
19. Ввести вектор дійсних чисел. Всі елементи більші за величину середнього арифметичного елементів вектора замінити на нуль, а максимальний елемент замінити на значення середнього арифметичного.
20. Задано з клавіатури відсортований за зростанням значень вектор дійсних чисел. Вставити в нього введене з клавіатури число так, щоб збереглась загальна впорядкованість елементів.
21. Ввести вектор дійсних чисел. Поміняти місцями сусідні елементи мінімального елемента, а максимальний елемент поділити на суму абсолютних значень від'ємних елементів.
22. З клавіатури ввести вектор дійсних чисел. Визначити значення і адреси двох найменших елементів введеного вектора та кількість нульових елементів.
23. З клавіатури ввести вектор дійсних чисел. Усі нульові елементи замінити на абсолютне значення суми від'ємних елементів.
24. З клавіатури ввести вектор додатних чисел. Сформувати новий вектор, елементи якого є середнім геометричним двох сусідніх елементів введеного вектора (1-го та 2-го елементів; 2-го та 3-го; 3-го та 4-го і т.д.).
25. Ввести вектор дійсних чисел. Всі елементи вектора, що розміщені перед його мінімальним елементом, зменшити на половину від середнього арифметичного елементів введеного вектора.

ДОДАТОК 2

Написати програму для обробки даних, організованих у масив, згідно завдання наведеного варіанту. У програмах використати різні форми звертання до елементів багатовимірних масивів, у тому числі за допомогою конструкцій “вказівник на масив” і “вказівник на вказівник”. Враховувати, що кожен рядок матриці можна опрацьовувати як окремий елемент.

1. Ввести прямокутну матрицю цілих чисел. Надрукувати номери рядків матриці, всі елементи яких непарні (або вивести повідомлення про відсутність таких рядків). Переставити елементи всіх інших рядків у зворотному порядку.
2. Ввести прямокутну матрицю дійсних чисел. Поміняти місцями два рядки матриці, з відповідно найбільшою і найменшою сумою елементів.
3. Ввести прямокутну матрицю цілих чисел. Циклічно зсунути її рядки на 2 вгору. Циклічний зсув передбачає, що елементи двох перших рядків матриці заносяться на відповідні місця зсунутих елементів двох останніх рядків.
4. Ввести квадратну матрицю дійсних чисел. Обчислити значення сум обох великих діагоналей матриці. Здійснити транспонування матриці.
5. Знайти найбільший елемент головної діагоналі матриці $A[5][5]$ і вивести на друк весь рядок, де він знаходиться.
6. Ввести прямокутну матрицю цілих чисел. Вилучити з матриці рядок і стовпчик, на перетині яких знаходиться її найменший елемент.
7. Для прямокутної матриці $X[5][7]$ обчислити скалярний добуток рядка, в якому знаходиться максимальний елемент матриці і стовпця, в якому знаходиться мінімальний елемент матриці.
8. Ввести 7 цілих беззнакових чисел, менших за 1000. Сформувати з цих чисел матрицю розмірністю 7×10 , кожен рядок якої заповнити **0** та **1** відповідно до двійкового коду введеного числа. Визначити число, двійковий код якого має найбільше **1**.
9. Ввести прямокутну матрицю дійсних чисел. Вилучити з матриці рядок, добуток елементів якого є найбільшим.
10. Пронормувати квадратну матрицю $A[5][5]$, розділивши всі елементи кожного рядка на максимальний за модулем елемент матриці.
11. Ввести прямокутну матрицю дійсних чисел. Циклічно зсунути її рядки на 1 униз. Циклічний зсув передбачає, що елементи останнього рядка матриці заносяться на відповідні місця зсунутих елементів першого рядка.
12. Ввести прямокутну матрицю дійсних чисел. Визначити номери двох рядків введеної матриці, скалярний добуток яких є найменшим.
13. Ввести прямокутну матрицю дійсних чисел. Вилучити з матриці рядок з найменшою сумою елементів.
14. Ввести прямокутну матрицю дійсних чисел. Зсунути її рядки на 2 вгору (два старших рядки матриці втрачаються). Звільнені останні рядки матриці заповнити **0**.
15. Обчислити суми елементів кожного рядка матриці $X[5][6]$, визначити найменше значення серед цих сум і номер відповідного рядка.
16. Задано квадратну матрицю $A[7][7]$. Перевірити чи матриця A симетрична відносно побічної діагоналі.
17. Ввести прямокутну матрицю дійсних чисел. Обчислити кількість нульових елементів в матриці. Рядок, в якому найбільше елементів замінити на перший рядок матриці.
18. Ввести прямокутну матрицю дійсних чисел. Обчислити кількість недодатніх елементів матриці. Стовпець, в якому найбільше недодатніх елементів замінити на останній стовпець матриці.
19. Для квадратної матриці цілих чисел B відсортувати у порядку спадання чисел кожний стовпець.
20. Ввести прямокутну матрицю дійсних чисел. Вилучити з матриці рядки з нульовою сумою елементів. Передбачити, що таких рядків може бути декілька.

21. Знайти найменші елементи кожного рядка квадратної матриці A і помістити їх на головну діагональ, а діагональні елементи на місця найбільших.
22. Ввести прямокутну матрицю дійсних чисел. Вилучити з матриці рядки, у яких найчастіше зустрічається комбінація від'ємного елемента по сусідству з додатнім.
23. Ввести прямокутну матрицю дійсних чисел. Вилучити з матриці рядок, для якого модуль різниці добутку елементів та їх суми є найбільшим.
24. Задано квадратну матрицю $A[5][5]$. Впорядкувати елементи головної діагоналі в порядку зростання.
25. `int A[15][15], B[15][15];` Знайти $C=AB$.