**ПРАКТИЧНА РОБОТА №3**

**РОБОТА З МАСИВАМИ**

**Мета роботи:** оволодіти практичними навиками по створенню одновимірних та багатовимірних масивів; знати особливості реалізації масивів на мові Java.

**Постановка задачі**

Масив являє собою область пам'яті, що містить фактичні дані, тобто значення елементів масиву. Змінна масиву фактично містить адресу цієї області пам'яті.

Для доступу до елементу масиву досить знати ім'я змінної масиву і індекс елемента в масиві.

Створення масиву в Java передбачаєреалізацію трьох етапів:

1. Створення (оголошення) змінної масиву.
2. Створення безпосередньо масиву - виділення в пам'яті місця для зберігання значень елементів масиву.
3. Присвоєння змінної масиву в якості значення посилання на масив. Змінна масиву отримує в якості значення адресу пам'яті, де зберігається масив.

Таким чином, змінна масив з технічної точки зору не є масив. Це змінна, яка «знає», де масив розташований.

**Створення змінної.**

В першу чергу створюємо змінну масиву ( «оголошуємо»). Для цього задаємо тип змінної і її ім'я. Щоб компілятор міг зрозуміти, що мова йде про змінну масиву, а не про звичайну змінну, після імені типу елементів масиву вказуються порожні квадратні дужки.

*Int [] Myarr;*

Це правило залишається справедливим і для масивів інших типів, причому не тільки базових. В Java існують посилальні типи. Змінні посилальних типів використовуються при роботі з об'єктами.

Оголошення змінної масиву зовсім не означає, що масив створений. Значенням змінної - є посилання. Але якщо масиву немає, то і посилатись нема на що. Фактично значення масиву дорівнює **null**.

Масив створюється незалежно від змінної масиву. Після створення масиву, змінній масиву необхідно в якості значення привласнити посилання на масив.

Обидва операції виконуються за допомогою оператора **new**. Його загальне призначення - виділення пам'яті в динамічному режимі.

Дія оператора **new** полягає в тому, що в пам'яті виділяється необхідна місце для зберігання даних. Як результат оператор **new** повертає посилання на область пам'яті, яка виділена.

При активізації оператора **new** йому потрібно вказати: для чого ця пам'ять виділяється і в якому обсязі. Робиться це в такий спосіб: після оператора **new** вказується тип елементів масиву, в квадратних дужках кількість елементів в масиві.

Приклад:

// Оголошення змінної масиву

**int[ ] myArr;**

// Створення масиву і запис посилання в змінну масиву

**myArr = new int[10];**

Якщо масив створюється таким чином, то всім елементам масиву автоматично присвоюються значення за замовчуванням. Наприклад, для числових значень початкове значення буде 0. Для масиву типу boolean початкове значення дорівнюватиме false, для масиву типу char - '\u0000', для масиву типу класу (об'єкти) - null.

На практиці команди оголошення змінної масиву і створення масиву зазвичай об'єднують в одну команду

**int [] myArr = new int [10];**

Звернення до елементу масиву виконується так: вказується ім'я масиву і в квадратних дужках індекс масиву.

Індексація елементів масиву в Java завжди починається **з нуля!**

Довжину масиву можна визначити за допомогою константи ***length***, яка визначається для кожного масиву

**int [] myArr = new int [10];**

**int len = meArr.length;**

*Ініціалізація масиву*

Можна ініціалізувати масив власними значеннями, коли він оголошується.

Для цього після оголошення змінної масиву слід додати знак рівності, за яким іде список значень елементів, розміщений у фігурні дужки. Ключове слово **new** не використовується:

**int [] cats = {2, 5, 7, 8, 3, 0}; // масив з 6 елементів**

*Копіювання масивів*

Масив - це не числа, а спеціальний об'єкт, який по особливому зберігається в пам'яті. Продемонструємо відміну масиву від простої змінної.

ПРИКЛАД. Припустимо, у нас є одна змінна, потім ми створили другу змінну і привласнили їй значення першої. А потім перевіримо їх вміст.

int a = 5;

int b = a;

System.out.print ("a =" + a + "\ nb =" + b);

Отримаємо очікуваний результат.

Спробуємо зробити подібне з масивом.

int[] anyNumbers = {2, 8, 11};

int[] luckyNumbers = anyNumbers;

luckyNumbers[2] = 25;

System.*out.print*("anyNumbers: “+Arrays.toString(anyNumbers) +"\nluckyNumbers: " + Arrays.toString(luckyNumbers));

Отримаємо результат.

**anyNumbers: [2, 8, 25]; luckyNumbers: [2, 8, 25];**

Якщо реально потрібна копія масиву, то використовують метод **Arrays.copyOf().**

Якщо програма вийде за межі індексу масиву, то програма зупиниться з помилкою часу виконання **ArrayOut Of Bounds Exception**. Це дуже часта помилка у програмістів.

**Багатовимірні масиви**

У Java можна створювати не тільки одномірні, але і багатовимірні масиви. В Java розмірність масиву може бути будь-яка, але на практиці масиви більшої розмірності, ніж два, використовується вкрай рідко.

З точки зору мови Java *двовимірний масив* - це одновимірний масив, елементами якого є одновимірні масиви.

Для *тривимірного масиву* - це одновимірний масив, елементами якого є двовимірні масиви. У Java можна створювати не тільки одномірні, але і багатовимірні масиви.

Як і в разі одновимірного масиву, слід розрізняти змінну двовимірного масиву і безпосередньо масив. При оголошенні змінної двовимірного масиву вказується тип елементів, двічі порожні квадратні дужки і ім'я змінної масиву.

**double[ ][ ] nA;**

При створенні двовимірного масиву використовується все той же оператор **new**. Після оператора **new** вказується тип елементів двовимірного масиву і розмірність по кожному з індексів.

**nA = new double[3][4];**

Команди оголошення змінної масиву і створення масиву можна об'єднати в одну:

**double[ ][ ] nA = new double[3][4];**

|  |  |  |  |
| --- | --- | --- | --- |
| **nA[0][0]** | **nA[0][1]** | **nA[0][2]** | **nA[0][3]** |
| **nA[1][0]** | **nA[1][1]** | **nA[1][2]** | **nA[1][3]** |
| **nA[2][0]** | **nA[2][1]** | **nA[2][2]** | **nA[2][3]** |

Не забуваємо, що індекси починаються з нуля.

В даному випадку ми визначили масив який складається з 3 рядків і 4 колонок.

Приклад створення двовимірного масиву з ініціалізацією власними значеннями:

**int [] [] a = {**

**{1, 2, 3},**

**{4, 5, 6}}**

При визначенні багатовимірного масиву можна вказати довжину тільки першого виміру. Довжини наступних вимірювань можна вказувати в будь-якому місці програми, в разі потреби.

int[][] twoD = new int[3][]; // память під перший вимір

// далі резервується пам’ять під другий вимір

twoD[0] = new int[4];

twoD[1] = new int[3];

twoD[2] = new int[2];

*Розмірність багатовимірного масиву*

Розмір двовимірного масиву вимірюється наступним чином. Довжина масиву визначається за його першої розмірності, тобто обчислюється кількість рядів.

int[][] matrix = new int[4][5];

System.out.println(matrix.length);

Повертає значення 4

А якщо потрібно дізнатися про кількість стовпців в ряду, слід вказати ряд, а потім обчислити кількість стовпців.

// число колонок у третього ряду

System.out.println(matrix[2].length);

*Зміна розмірів масиву*

Розмір масиву змінити НЕ МОЖНА, але можна створити новий масив, скопіювати в нього елементи з старого і продовжувати використовувати ту ж саму змінну:

public class Ara{

public static void main(String[] args) {

int[] elements = {1, 2, 3, 4, 5}; //массив из 5 элементов

int[] tmp = new int[10];

System.arraycopy(elements, 0, tmp, 0, elements.length);

elements = tmp;

for(int i:elements) {

System.out.println(i);

}

}}

В цьому прикладі був застосовано метод **arraycopy** з класу **System**. Метод **arraycopy** має синтаксис

***static void arraycopy(Object джерело, int початокДжерела,  Objeсt мета,int початокМети, int розмір)***

**arraycopy -**метод копіювання масиву. Масив, що підлягає копіюванню, передається в параметрі *джерело*, а індекс позиції, з якої починається копіювання з джерела, - в *початокДжерела*. Масив, що приймає копію, *Object мета*, а індекс позиції в ньому, куди потрібно почати копіювання, - в параметрі *початокМети*. Параметр *розмір* задає кількість копійованих елементів.

*Використання циклів для обробки масивів*

Починаючи з версії JDK 5 в Java можна використовувати другу форму циклу **for**, що реалізує цикл в стилі "**for-each**" («для кожного»).

Це більш компактна форма **for** для перебору елементів масивів - **foreach**. Конструкція **foreach** не вимагає ручного зміни змінної, кроку для перебору, цикл автоматично виконує цю роботу.

Цикл **foreach**, має наступний синтаксис:

**for(*тип ітер\_пер : коллекция*) *блок\_операторов***

Тут *тип* вказує тип, а *ітер\_пер* - ім'я ітераційної змінної, яка послідовно буде приймати значення з *колекції*, від першого до останнього. Елемент *колекція* вказує *колекцію*, по якій здійснюватиметься цикл. На кожній ітерації циклу програма витягує наступний елемент колекції і зберігає його у змінній *ітер\_пер*. Цикл буде виконуватися до тих пір, поки не будуть отримані всі елементи колекції.

Для порівняння напишемо цикл для обчислення суми значень елементів масиву традиційним способом:

int[] nums = { 1, 2, 3, 4, 5 };

int sum = 0;

for(int i = 0; i < 5; i++) sum += nums[i];

Цей код можна переписати таким чином:

int[] nums = { 1, 2, 3, 4, 5 };

int sum = 0;

for(int i : nums) sum += i;

При проходженні циклу, змінній *і* автоматично присвоюється значення, яке дорівнює наступному елементу масива *nums*. Таким чином, при першому проходженні змінна *i* містить значення 1, при другому - 2 і т.д. Крім того при такому способі виключається можливість появи помилки - вихід за межі масиву.

Оскільки кожен оператор **for** в стилі **foreach** перебирає елементи масиву послідовно, починаючи з першого і закінчуючи останнім, то даний спосіб зручний для багатьох операцій. Наприклад, для пошуку значення в неупорядкованому масиві. Пошук припиняється після виявлення потрібного значення.

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

int val = 5;

boolean found = false;

// шукаємо значення 5 в масиві

for (intx : nums) {

if (x == val) {

found = true;

break;

} }

if (found) {

System.out.println("Значення знайдено");

}

**ПРИКЛАДИ ВИКОНАННЯ ІНДИВІДУАЛЬНИХ ЗАДАНЬ**

**Скласти два масиви**

Припустимо, є два масиви, і потрібно їх поєднати і отримати загальний масив.

private double[] concatArray(double[] a, double[] b) {

if (a == null)

return b;

if (b == null)

return a;

double[] r = new double[a.length + b.length];

System.arraycopy(a, 0, r, 0, a.length);

System.arraycopy(b, 0, r, a.length, b.length);

return r;

}

Аналогічно, якщо потрібно взяти тільки частину з великого масиву, то можна скористатися методом:

// start – з якої позиції потрібно отримати новий масив, відлік

// починається з 0

private double[] copyPartArray(double[] a, int start) {

if (a == null)

return null;

if (start > a.length)

return null;

double[] r = new double[a.length - start];

System.arraycopy(a, start, r, 0, a.length - start);

return r; }

У наступній програмі створюються і ініціалізуються масиви та

обчислення добутку двох матриць.

/\* приклад: розрахунок добутку двох матриць:

Matrix.java \*/

public class Matrix {

private int[][] a;

// Створення і заповнення матриці випадковими значеннями

Matrix(int n, int m) {

//

a = new int[n][m];

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

for (int j = 0; j < m; j++)

a[i][j] = (int) (Math.random() \* 5);

show();

}

// Створення і заповнення матриці значеннями k

public Matrix(int n, int m, int k) {

a = new int[n][m];

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

for (int j = 0; j < m; j++) {

a[i][j] = k;

}

show();

if(k!=0) show();

}

// вивід матриці на консоль

**public void** show() {

System.***out***.println("Maтриця : " + a.length + " на " + a[0].length);

**for** (**int** i = 0; i < a.length; i++) {

**for** (**int** j = 0; j < a[0].length; j++)

System.***out***.print(a[i][j] + " ");

System.***out***.println();

}

}

**public static void** main(String[] args) {

**int** n = 2, m = 3, l = 4;

Matrix p = **new** Matrix(n, m);

Matrix q = **new** Matrix(m, l);

Matrix r = **new** Matrix(n, l, 0);

// обчислення добутку двох матриць

**for** (**int** i = 0; i < p.a.length; i++)

**for** (**int** j = 0; j < q.a[0].length; j++)

**for** (**int** k = 0; k < p.a[0].length; k++)

r.a[i][j] += p.a[i][k] \* q.a[k][j];

System.***out***.println("Добутокматриць: ");

r.show();

}

}

Формули для розрахунку добутку:





Результат виконання програми:

Maтрица : 2 на 3

3 0 4

1 4 0

Maтрица : 3 на 4

4 1 3 1

1 3 0 0

4 0 1 3

Maтрица : 2 на 4

0 0 0 0

0 0 0 0

Произведение матриц:

Maтрица : 2 на 4

28 3 13 15

8 13 3 1

**Індивідуальні завдання**

**Завдання 1**

|  |  |
| --- | --- |
| **№ варіанту** | **Задачі** |
| 1, 6, 11, 16, 21, 26, 31 | 1. Напишіть програму, що заміняє нулями всі від’ємні елементи заданого не порожнього масиву, що передують його першому максимальному елементу. 2. Дано масив . Підрахувати найбільшу кількість підряд розміщених однакових елементів. 3. В одномірному масиві, що складається з дійсних елементів, обчислити суму від’ємних елементів масиву; добуток елементів масиву, розташованих між максимальним і мінімальним елементами. Упорядкувати елементи масиву за зростанням. 4. В одновимірному масиві, що складається з цілих елементів, обчислити мінімальний за модулем елемент масиву; суму модулів елементів масиву, розташованих після першого елемента, рівного нулю. Перетворити масив таким чином, щоб у першій його половині розташовувалися елементи, що стояли на парних позиціях, а в другій половині − елементи, що стояли на непарних позиціях. 5. В одновимірному масиві, що складається з дійсних елементів, обчислити кількість елементів масиву, менших введеного з клавіатури числа ; суму цілих частин елементів масиву, розташованих після останнього від’ємного елемента. Перетворити масив таким чином, щоб спочатку розташовувалися всі елементи, що відрізняються від максимального не більше ніж на 20%, а потім − всі інші. 6. В одновимірному масиві, що складається з дійсних елементів, обчислити кількість від’ємних елементів масиву; суму модулів елементів масиву, розташованих після мінімального за модулем елемента.   Замінити всі від’ємні елементи масиву їхніми квадратами й упорядкувати елементи масиву за зростанням. |
| 2, 7, 12, 19, 24, 29, 34 | 1. Напишіть програму, що змінює місцями найбільший і найменший елементи заданого не порожнього масиву різних чисел і потім друкує отриманий масив. 2. В масиві кожний елемент рівний 0, 1 або 2. Переставити елементи масиву так, щоб спочатку розташовувались всі нулі, потім всі одиниці, а потім всі двійки (додатковий масив не використовувати). 3. В одномірному масиві, що складається з дійсних елементів, обчислити суму додатних елементів масиву; добуток елементів масиву, розташованих між максимальним за модулем і мінімальним за модулем елементами. Упорядкувати елементи масиву за спаданням. 4. В одновимірному масиві, що складається з дійсних елементів, обчислити номер мінімального за модулем елемента масиву; суму модулів елементів масиву, розташованих після першого від’ємного елемента. Стиснути масив, видаливши з нього всі елементи, величина яких знаходиться в інтервалі . Елементи, що звільнилися в кінці масиву, заповнити нулями. 5. В одномірному масиві, що складається з дійсних елементів, обчислити номер максимального за модулем елемента масиву; суму елементів масиву, розташованих після першого додатного елемента. Перетворити масив таким чином, щоб спочатку розташовувалися всі елементи, ціла частина яких лежить в інтервалі , а потім-всі інші. 6. В одновимірному масиві, що складається з дійсних елементів, обчислити кількість елементів масиву, які більші за 3; добуток елементів масиву, розташованих після максимального за модулем елемента.   Перетворити масив таким чином, щоб спочатку розташовувалися всі від’ємні елементи, а потім − усі додатні (елементи, рівні 0, вважати додатними). |
| 3, 8, 13, 17, 22, 27, 32 | 1. Напишіть програму, що вводить із клавіатури число *A* і друкує кількість елементів заданого масиву, більших за *A*. 2. Задано числовий масив . Вирахувати та надрукувати кількість різних чисел в цьому масиві. Наприклад в масиві 5,7,5 є 2 різних числа 7 і 5. 3. В одновимірному масиві, що складається з цілих елементів, обчислити: добуток елементів масиву з парними номерами; суму елементів масиву, розташованих між першим і останнім нульовим елементами. Перетворити масив таким чином, щоб спочатку розташовувалися всі додатні елементи, а потім − усі від’ємні (елементи, рівні 0, вважати додатними). 4. В одновимірному масиві, що складається з дійсних елементів, обчислити: суму елементів масиву з непарними номерами; суму елементів масиву, розташованих між першим і останнім від’ємними елементами. Стиснути масив, видаливши з нього всі елементи, модуль яких не перевищує 1. Елементи, що звільнилися в кінці масиву, заповнити нулями. 5. В одновимірному масиві, що складається з дійсних елементів, обчислити максимальний за модулем елемент масиву; суму елементів масиву, розташованих між першим і другим додатними елементами. Перетворити масив таким чином, щоб елементи, рівні нулю, розташовувалися після всіх інших. 6. В одновимірному масиві, що складається з дійсних елементів, обчислити кількість елементів масиву, рівних 0; суму елементів масиву, розташованих після мінімального елемента. Упорядкувати елементи масиву за зростанням модулів елементів. |
| 4, 9, 14, 18, 23, 28, 33 | 1. Напишіть програму, що друкує другий по величині елемент заданого не порожнього масиву різних чисел. 2. Задати одновимірний масив із цілих чисел, що дорівнюють 0, 1, 2, m=15. Поміняти числа місцями таким чином, щоб вони розташувались у такому порядку: 0, 1, 2. 3. В одномірному масиві, що складається з дійсних елементів, обчислити максимальний елемент масиву; суму елементів масиву, розташованих до останнього додатного елемента. Стиснути масив, видаливши з нього всі елементи, модуль яких знаходиться в інтервалі . Елементи, що звільнилися в кінці масиву, заповнити нулями. 4. В одновимірному масиві, що складається з дійсних елементів, обчислити номер мінімального елемента масиву; суму елементів масиву, розташованих між першим і другим від’ємними елементами. Перетворити масив таким чином, щоб спочатку розташовувалися всі елементи, модуль яких не перевищує 1, а потім − всі інші. 5. В одновимірному масиві, що складається з дійсних елементів, обчислити добуток від’ємних елементів масиву; суму додатних елементів масиву, розташованих до максимального елемента. Змінити порядок проходження елементів у масиві на зворотний. 6. В одновимірному масиві, що складається з цілих елементів, обчислити кількість додатних елементів масиву; суму елементів масиву, розташованих після останнього елемента, рівного нулю. Перетворити масив таким чином, щоб спочатку розташовувалися всі елементи, ціла частина яких не перевищує 1, а потім − всі інші. |
| 5, 10, 15, 20, 25, 30, 35 | 1. Напишіть програму, що друкує номер першого нульового елемента заданого масиву чи повідомлення про те, що масив не містить нульових елементів. 2. Напишіть програму, що заміняє всі елементи заданого масиву, крім крайніх, на півсуму сусідніх, і друкує результат. 3. В одновимірному масиві, що складається з дійсних елементів, обчислити мінімальний елемент масиву; суму елементів масиву, розташованих між першим і останнім додатними елементами. Перетворити масив таким чином, щоб спочатку розташовувалися всі елементи рівні нулю, а потім − всі інші. 4. В одновимірному масиві, що складається з цілих елементів, обчислити номер максимального елемента масиву; добуток елементів масиву, розташованих між першим і другим нульовими елементами. Перетворити масив таким чином, щоб у першій його половині розташовувалися елементи, що стояли на непарних позиціях, а в другій половині − елементи, що стояли на парних позиціях. 5. В одновимірному масиві, що складається з дійсних елементів, обчислити добуток додатних елементів масиву; суму елементів масиву, розташованих до мінімального елемента. Упорядкувати за зростанням окремо елементи, що розташовані на парних місцях, і елементи, що розташовані на непарних місцях. 6. В одновимірному масиві, що складається з дійсних елементів, обчислити кількість елементів масиву, що лежать у діапазоні від до ; суму елементів масиву, розташованих після максимального елемента. Упорядкувати елементи масиву за спаданням модулів елементів. |

**Завдання 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ варіанту** | **Технологічна карта** | | |
|  | Розмірність масиву | Діапазон значень | Завдання |
| 1 | 200 | -50..50 | Обчислити кількість пар сусідніх елементів з однаковими значеннями. |
| 2 | 150 | -100..100 | Обчислити кількість ділянок, які утворюють неперервні послідовності чисел з не спадаючими значеннями. |
| 3 | 200 | -150..100 | Знайти неперервну послідовність додатних чисел, сума елементів в якій – максимальна. |
| 4 | 100 | -100..100 | Розташувати всі додатні числа в лівій частині, від’ємні – в правій і «0» - по центру між ними. |
| 5 | 200 | 0..100 | Знайти значення n-го за величиною елемента і значення всіх елементів масиву, які його перевищують, замінити на знайдене значення. Вивести вхідний, вихідний масиви та значення n-го елемента. |
| 6 | 200 | -50..50 | Вивести довжину і початкові індекси всіх неперервних послідовностей чисел, які утворюють зростаючу послідовність. |
| 7 | 300 | -100..150 | Вивести довжину і початкові індекси всіх послідовностей, що містять знакозмінний ряд чисел. |
| 8 | 100 | -50..50 | Знайти з неперервних послідовностей від’ємних чисел ту послідовність, яка має найбільше абсолютне значення середнього арифметичного її значення. |
| 9 | 200 | 0..100 | Знайти неперервну ділянку із 10 елементів, сума яких максимальна. |
| 10 | 300 | -100…100 | Знайти неперервні ділянки, на яких сума елементів дорівнює 0. |
| 11 | 100 | -50..50 | Визначити, яке абсолютне значення в масиві зустрічається найчастіше. |
| 12 | 120 | -70..70 | Визначити середню довжину неперервних додатних послідовностей масиву (проміжні значення теж виводити). |
| 13 | 300 | -200..200 | Для всіх послідовностей від’ємних чисел поміняти місцями елементи з максимальним та мінімальним значенням. |
| 14 | 100 | -50..50 | Вивести довжину і початкові індекси всіх неперервних послідовностей чисел, які утворюють спадну послідовність. |
| 15 | 100 | -100..100 | Обчислити кількість пар сусідніх елементів з різними значеннями. |
| 16 | 150 | -100..100 | Обчислити кількість ділянок, які утворюють неперервні послідовності чисел з зростаючими значеннями. |
| 17 | 100 | -150..100 | Знайти неперервну послідовність додатних чисел, сума елементів в якій – мінімальна. |
| 18 | 200 | 0..100 | Знайти значення n-го за величиною елемента і значення всіх елементів масиву, які менші за нього, замінити на знайдене значення. Вивести вхідний, вихідний масиви та значення n-го елемента. |
| 19 | 100 | -50..50 | Знайти з неперервних послідовностей додатних чисел ту послідовність, яка має найменше значення середнього арифметичного її значення. |
| 20 | 100 | -50..50 | Обчислити кількість пар сусідніх елементів з однаковими знаками. |
| 21 | 200 | -50..50 | Обчислити кількість пар сусідніх елементів з однаковими значеннями. |
| 22 | 150 | -100..100 | Обчислити кількість ділянок, які утворюють неперервні послідовності чисел з не спадаючими значеннями. |
| 23 | 200 | -150..100 | Знайти неперервну послідовність додатних чисел, сума елементів в якій – максимальна. |
| 24 | 100 | -100..100 | Розташувати всі додатні числа в лівій частині, від’ємні – в правій і «0» - по центру між ними. |
| 25 | 200 | 0..100 | Знайти значення n-го за величиною елемента і значення всіх елементів масиву, які його перевищують, замінити на знайдене значення. Вивести вхідний, вихідний масиви та значення n-го елемента. |
| 26 | 200 | -50..50 | Вивести довжину і початкові індекси всіх неперервних послідовностей чисел, які утворюють зростаючу послідовність. |
| 27 | 300 | -100..150 | Вивести довжину і початкові індекси всіх послідовностей, що містять знакозмінний ряд чисел. |
| 28 | 100 | -50..50 | Знайти з неперервних послідовностей від’ємних чисел ту послідовність, яка має найбільше абсолютне значення середнього арифметичного її значення. |
| 29 | 200 | 0..100 | Знайти неперервну ділянку із 10 елементів, сума яких максимальна. |
| 30 | 300 | -100…100 | Знайти неперервні ділянки, на яких сума елементів дорівнює 0. |
| 31 | 100 | -50..50 | Визначити, яке абсолютне значення в масиві зустрічається найчастіше. |
| 32 | 120 | -70..70 | Визначити середню довжину неперервних додатних послідовностей масиву (проміжні значення теж виводити). |
| 33 | 300 | -200..200 | Для всіх послідовностей від’ємних чисел поміняти місцями елементи з максимальним та мінімальним значенням. |
| 34 | 100 | -50..50 | Вивести довжину і початкові індекси всіх неперервних послідовностей чисел, які утворюють спадну послідовність. |
| 35 | 100 | -100..100 | Обчислити кількість пар сусідніх елементів з різними значеннями. |

**Завдання 3**

|  |  |
| --- | --- |
| **№ варіанту** | **Задача** |
| 1 | 1. Визначити мінімальний елемент кожного рядка матриці. Надрукувати його значення та індекси. 2. Знайти добуток додатних елементів квадратної матриці, які розміщені на її головній діагоналі. 3. Ввести двовимірний масив, у якому числа 0 і 1 розміщені випадковим чином, а число стовпців не збігається із числом рядків. Визначити функцію ініціалізації масиву й функцію виведення на екран. Порахувати скільки сусідніх елементів, рівних 1, є у елемента (2;2). |
| 2 | 1. Для кожного стовпця матриці визначити і надрукувати суму значень елементів, розміщених в рядках з непарними номерами. 2. Знайти суму від’ємних елементів квадратної матриці, які розміщені на її головній діагоналі. 3. На головній діагоналі квадратной матрицірозташуватиі числа від N до 1, під головною діагоналлю нулі, а над головною діагоналлю по рядках числа в порядку зростання від заданого. Оформити заповнення масиву у вигляді окремої функції ініціалізації. Вивести отриману матрицю за допомогою функції. |
| 3 | 1. Визначити і надрукувати кількість додатних елементів кожного стовпця матриці. 2. Знайти суму додатних елементів квадратної матриці, які розміщені на її головній діагоналі. 3. На головній діагоналі квадратной матрицізнаходяться елементи рівні 1, нижче головної діагоналі - 0, а вище - сумі індексів. Оформити заповнення масиву у вигляді окремої функції. Виведіть матрицю на екран (функція). |
| 4 | 1. Для кожного рядка матриці надрукувати номери стовпців, що містять додатні елементи. 2. Знайти добуток від’ємних елементів квадратної матриці, які розміщені на її головній діагоналі. 3. Написати функцію транспонування квадратної матриці. З її допомогою визначити чи є задана матриця симетричної (транспонована матриця дорівнює вихідній). |
| 5 | 1. Для кожного рядка матриці визначити і надрукувати кількість елементів, значення яких перевищує число Z. 2. Знайти і надрукувати добуток значень від’ємних елементів матриці. |
| 6 | 1. Знайти і надрукувати суму значень додатних елементів матриці. 2. Для кожного стовпця матриці визначити і надрукувати кількість елементів, значення яких не перевищує число Z. 3. Заповнити квадратну матрицю випадковими числами (функція). Написати функцію транспонування матриці. Написати функцію додавання матриць. Скласти вихідну матрицю й транспоновану. Вивести матриці на екран (функція). |
| 7 | 1. Знайти максимальний по модулю елемент матриці. Надрукувати його значення та індекси. 2. Для кожного рядка матриці знайти і надрукувати суму значень елементів, що розміщені в стовпцях з непарними номерами. |
| 8 | 1. Визначити і надрукувати кількість від’ємних елементів у кожному стовпці матриці. 2. Визначити і надрукувати найменший елемент матриці та його індекси. 3. Заповнити квадратну матрицю випадковими числами (функція). Написати функцію для видалення одного рядка. Після видалення рядка останній рядок матриці повинен бути заповнено нулями. Вивести вихідну й отриману матриці на екран (функція) |
| 9 | 1. Для кожного стовпця матриці визначити і надрукувати кількість елементів, значення яких перевищує число Z. 2. Знайти і надрукувати суму значень від’ємних елементів матриці. 3. Написати функцію для обчислення суми елементів квадратної матриці, які розташовані нижче головної діагоналі. З її допомогою знайти максимальне значення такої суми (функція) в N матрицях випадкових чисел (функція). |
| 10 | 1. Для кожного рядка матриці знайти і надрукувати суму значень елементів, розміщених в стовпцях з парними номерами. 2. Знайти і надрукувати кількість значень від’ємних елементів матриці. 3. Заповнити квадратну матрицю випадковими числами (функція). Написати функцію для пошуку мінімального елемента в зазначеному рядку. Зрушити елементи цього рядка циклічно вліво на кількість елементів, рівну мінімальному елементу рядка. Вивести вихідну й отриману матриці (функція). |
| 11 | 1. Визначити і надрукувати кількість додатних елементів матриці. 2. Для кожного рядка матриці визначити і надрукувати номери стовпців, що містять від’ємні елементи. 3. Заповнити квадратну матрицю випадковими числами (функція). Знайти мінімальний з неповторюваних елементів матриці. Вивести матрицю (функція) і мінімальний елемент на екран. |
| 12 | 1. Для кожного рядка матриці знайти і надрукувати суму значень від’ємних елементів. 2. Знайти мінімальний по модулю елемент матриці. 3. Заповнити квадратну матрицю випадковими числами (функція). Знайти середнє арифметичне першого стовпця (функція) і кількість елементів матриці, що перевищують середнє арифметичне першого стовпця. |
| 13 | 1. Для кожного стовпця матриці знайти і надрукувати номери рядків, що містять від’ємні елементи. 2. Знайти добуток елементів матриці, що знаходяться на головній діагоналі матриці. 3. Заповнити квадратну матрицю випадковими числами (функція). Знайти середнє арифметичне головної діагоналі (функція) і кількість елементів у першому рядку, менших цього середнього арифметичного. |
| 14 | 1. Для кожного рядка матриці знайти і надрукувати кількість додатних елементів. 2. Визначити максимальний по модулю елемент матриці. 3. Заповнити квадратну матрицю випадковими числами (функція). Знайти середнє арифметичне найбільшого (функція) і найменшого (функція) її елементів. Замінити отриманим середнім арифметичним всі елементи заданого рядка. Вивести на екран вихідний і отриманий масиви (функція). |
| 15 | 1. Для кожного рядка матриці знайти і надрукувати максимальний елемент. 2. Знайти кількість від’ємних елементів матриці. 3. Заповнити двовимірний масив випадковими цілими числами (функція). Знайти найбільший (функція) і найменший (функція) елементи масиву й, чергуючи, заповнити ними одновимірний масив заданої розмірності. Вивести вихідний (функція) і одновимірний (функція) масиви на екран. |
| 16 | 1. Визначити мінімальний елемент кожного рядка матриці. Надрукувати його значення та індекси. 2. Знайти добуток додатних елементів квадратної матриці, які розміщені на її головній діагоналі.   Ввести двовимірний масив, у якому числа 0 і 1 розміщені випадковим чином, а число стовпців не збігається із числом рядків. Визначити функцію ініціалізації масиву й функцію виведення на екран. Порахувати скільки сусідніх елементів, рівних 1, є у елемента (2;2). |
| 17 | 1. Для кожного стовпця матриці визначити і надрукувати суму значень елементів, розміщених в рядках з непарними номерами. 2. Знайти суму від’ємних елементів квадратної матриці, які розміщені на її головній діагоналі.   На головній діагоналі квадратной матрицірозташуватиі числа від N до 1, під головною діагоналлю нулі, а над головною діагоналлю по рядках числа в порядку зростання від заданого. Оформити заповнення масиву у вигляді окремої функції ініціалізації. Вивести отриману матрицю за допомогою функції. |
| 18 | 1. Визначити і надрукувати кількість додатних елементів кожного стовпця матриці. 2. Знайти суму додатних елементів квадратної матриці, які розміщені на її головній діагоналі.   На головній діагоналі квадратной матрицізнаходяться елементи рівні 1, нижче головної діагоналі - 0, а вище - сумі індексів. Оформити заповнення масиву у вигляді окремої функції. Виведіть матрицю на екран (функція). |
| 19 | 1. Для кожного рядка матриці надрукувати номери стовпців, що містять додатні елементи. 2. Знайти добуток від’ємних елементів квадратної матриці, які розміщені на її головній діагоналі.   Написати функцію транспонування квадратної матриці. З її допомогою визначити чи є задана матриця симетричної (транспонована матриця дорівнює вихідній). |
| 20 | 1. Для кожного рядка матриці визначити і надрукувати кількість елементів, значення яких перевищує число Z.   Знайти і надрукувати добуток значень від’ємних елементів матриці. |
| 21 | 1. Знайти і надрукувати суму значень додатних елементів матриці. 2. Для кожного стовпця матриці визначити і надрукувати кількість елементів, значення яких не перевищує число Z.   Заповнити квадратну матрицю випадковими числами (функція). Написати функцію транспонування матриці. Написати функцію додавання матриць. Скласти вихідну матрицю й транспоновану. Вивести матриці на екран (функція). |
| 22 | 1. Знайти максимальний по модулю елемент матриці. Надрукувати його значення та індекси.   Для кожного рядка матриці знайти і надрукувати суму значень елементів, що розміщені в стовпцях з непарними номерами. |
| 23 | 1. Визначити і надрукувати кількість від’ємних елементів у кожному стовпці матриці. 2. Визначити і надрукувати найменший елемент матриці та його індекси.   Заповнити квадратну матрицю випадковими числами (функція). Написати функцію для видалення одного рядка. Після видалення рядка останній рядок матриці повинен бути заповнено нулями. Вивести вихідну й отриману матриці на екран (функція) |
| 24 | 1. Для кожного стовпця матриці визначити і надрукувати кількість елементів, значення яких перевищує число Z. 2. Знайти і надрукувати суму значень від’ємних елементів матриці.   Написати функцію для обчислення суми елементів квадратної матриці, які розташовані нижче головної діагоналі. З її допомогою знайти максимальне значення такої суми (функція) в N матрицях випадкових чисел (функція). |
| 25 | 1. Для кожного рядка матриці знайти і надрукувати суму значень елементів, розміщених в стовпцях з парними номерами. 2. Знайти і надрукувати кількість значень від’ємних елементів матриці.   Заповнити квадратну матрицю випадковими числами (функція). Написати функцію для пошуку мінімального елемента в зазначеному рядку. Зрушити елементи цього рядка циклічно вліво на кількість елементів, рівну мінімальному елементу рядка. Вивести вихідну й отриману матриці (функція). |
| 26 | 1. Визначити і надрукувати кількість додатних елементів матриці. 2. Для кожного рядка матриці визначити і надрукувати номери стовпців, що містять від’ємні елементи.   Заповнити квадратну матрицю випадковими числами (функція). Знайти мінімальний з неповторюваних елементів матриці. Вивести матрицю (функція) і мінімальний елемент на екран. |
| 27 | 1. Для кожного рядка матриці знайти і надрукувати суму значень від’ємних елементів. 2. Знайти мінімальний по модулю елемент матриці.   Заповнити квадратну матрицю випадковими числами (функція). Знайти середнє арифметичне першого стовпця (функція) і кількість елементів матриці, що перевищують середнє арифметичне першого стовпця. |
| 28 | 1. Для кожного стовпця матриці знайти і надрукувати номери рядків, що містять від’ємні елементи. 2. Знайти добуток елементів матриці, що знаходяться на головній діагоналі матриці.   Заповнити квадратну матрицю випадковими числами (функція). Знайти середнє арифметичне головної діагоналі (функція) і кількість елементів у першому рядку, менших цього середнього арифметичного. |
| 29 | 1. Для кожного рядка матриці знайти і надрукувати кількість додатних елементів. 2. Визначити максимальний по модулю елемент матриці.   Заповнити квадратну матрицю випадковими числами (функція). Знайти середнє арифметичне найбільшого (функція) і найменшого (функція) її елементів. Замінити отриманим середнім арифметичним всі елементи заданого рядка. Вивести на екран вихідний і отриманий масиви (функція). |
| 30 | 1. Для кожного рядка матриці знайти і надрукувати максимальний елемент. 2. Знайти кількість від’ємних елементів матриці.   Заповнити двовимірний масив випадковими цілими числами (функція). Знайти найбільший (функція) і найменший (функція) елементи масиву й, чергуючи, заповнити ними одновимірний масив заданої розмірності. Вивести вихідний (функція) і одновимірний (функція) масиви на екран. |
| 31 | 1. Визначити мінімальний елемент кожного рядка матриці. Надрукувати його значення та індекси. 2. Знайти добуток додатних елементів квадратної матриці, які розміщені на її головній діагоналі.   Ввести двовимірний масив, у якому числа 0 і 1 розміщені випадковим чином, а число стовпців не збігається із числом рядків. Визначити функцію ініціалізації масиву й функцію виведення на екран. Порахувати скільки сусідніх елементів, рівних 1, є у елемента (2;2). |
| 32 | 1. Для кожного стовпця матриці визначити і надрукувати суму значень елементів, розміщених в рядках з непарними номерами. 2. Знайти суму від’ємних елементів квадратної матриці, які розміщені на її головній діагоналі.   На головній діагоналі квадратной матрицірозташуватиі числа від N до 1, під головною діагоналлю нулі, а над головною діагоналлю по рядках числа в порядку зростання від заданого. Оформити заповнення масиву у вигляді окремої функції ініціалізації. Вивести отриману матрицю за допомогою функції. |
| 33 | 1. Визначити і надрукувати кількість додатних елементів кожного стовпця матриці. 2. Знайти суму додатних елементів квадратної матриці, які розміщені на її головній діагоналі.   На головній діагоналі квадратной матрицізнаходяться елементи рівні 1, нижче головної діагоналі - 0, а вище - сумі індексів. Оформити заповнення масиву у вигляді окремої функції. Виведіть матрицю на екран (функція). |
| 34 | 1. Для кожного рядка матриці надрукувати номери стовпців, що містять додатні елементи. 2. Знайти добуток від’ємних елементів квадратної матриці, які розміщені на її головній діагоналі.   Написати функцію транспонування квадратної матриці. З її допомогою визначити чи є задана матриця симетричної (транспонована матриця дорівнює вихідній). |
| 35 | 1. Для кожного рядка матриці визначити і надрукувати кількість елементів, значення яких перевищує число Z.   Знайти і надрукувати добуток значень від’ємних елементів матриці. |