

Лабораторна робота № 7. Масиви. Робота з вектором і матрицею.



Масив - це сукупність змінних, які мають загальне ім'я й базовий тип. Усі елементи даних, що зберігаються в масиві, повинні мати той самий тип. Інформація, збережена в масиві, може бути доступна в будь-якому порядку.

Масив дозволяє зберігати й маніпулювати багатьма елементами даних за допомогою єдиної змінної. Обробку масивів значно спрощує використання циклів.

Найпростіший варіант масиву це одномірний масив, що використовує звичайний список даних. Наприклад:

Вася, Петро, Микола, Михайло, Іван, Слава, Ігор, Юра

Це строковий масив, що полягає з 8 елементів. Дамо йому назву *Name_Array*. Нумерація елементів у масиві починається з 0. Така система нумерації досить поширена в програмуванні й називається *нумерацією з нульовою базою*.

Для доступу до даних, що зберігаються в певному елементі масиву, слід вказувати ім'я масиву з наступним числом, яке має назву індекса елемента. Індекс завжди записується у круглій дужки. Наприклад: *Name_Array(3)* - цьому елементу масиву відповідає "*Михайло*" (не забувайте, що за замовчуванням нумерація елементів масиву починається з 0).

Оскільки нумерація з нульовою базою не дуже зручна (тому що ми звикли рахувати з 1, а не з 0), то в VBA є директива компілятора, що дозволяє виправити цю "незручність": *Option Base*.

Директива компілятора має два варіанти написання:

Option Base 0 - індекси масивів починаються з 0 (установка за замовчуванням)

Option Base 1 - індекси масивів починаються з 1

Дана директива компілятора розміщується в область оголошень модуля перед оголошеннями будь-яких змінних, констант або процедур. Не можна поміщати *Option Base* усередині процедури. Оператор *Option Base* може бути тільки один і він впливає на всі масиви, що оголошуються в модулі.

Одномірні масиви добре підходять для організації простих списків даних. Однак, часто буває необхідно представити таблиці даних у програмах з організацією даних у форматі рядків і стовпців, подібно коміркам у робочих аркушах *Excel*. Для цього необхідно використовувати багатомірні масиви. Так адреса кожної комірки аркуша складається із двох чисел, одне з яких (номер рядка) є першим індексом, а друге (номер стовпця) - другим індексом масиву. Такий масив називається *двовимірним* масивом. Додавши ще номер аркуша,

одержимо *тривимірний* масив. VBA дозволяє створювати масиви, що мають до 60 вимірів.

Масиви, що не міняють число своїх елементів, називаються *статичними* масивами. Прикладом такого масиву може служити вищенаведений масив *Name_Array*, що містить 8 елементів.

Однак, бувають ситуації, коли кількість елементів у масиві спочатку невідома, або ж, у процесі роботи ця кількість може змінюватися. Такі масиви називаються *динамічними* масивами.

Динамічний масив може збільшуватися або зменшуватися, щоб вміщати точно необхідне число елементів без марної витрати пам'яті.

Оголошення масиву, як і оголошення змінної, здійснюється з використанням оператора **Dim**:

Dim Varname ([Subscripts]) [**As** Type],

де *Varname* - будь-яке ім'я масиву, що використовує припустимий ідентифікатор імені;

Subscripts - вимір масиву. Якщо розмірність масиву більше одиниці, то *Subscripts* розділяється комами.

Оператор *Subscripts* має наступний синтаксис:

[lower **To**] upper [, [lower **To**] upper].

Необов'язковий аргумент *lower* визначає нижній діапазон припустимих індексів для масиву, а обов'язковий аргумент *upper* – верхню межу для індексів масиву.

Приклади правильного оголошення масивів:

- одномірний статичний строковий масив, що включає 10 елементів:

Dim Array_Str(1 **To** 10) **As** String;

- динамічний масив:

Dim Array_Var();

- двовимірний статичний масив цілих чисел, що включає 6*8=48 елементів :

Dim Array_Mult(0 **To** 5, 0 **To** 7) **As** Integer.

При оголошенні масивів слід пам'ятати, що включення оператора *Subscripts* в оголошенні масиву створює статичний масив з фіксованим числом елементів, пропуск оператора *Subscripts* в оголошенні масиву створює динамічний масив. Установка директиви компілятора *Option Base* впливає на загальне число елементів у масиві.

При використанні динамічних масивів можуть виникнути обставини, при яких точно невідомо, скільки елементів буде потрібно в масиві. В VBA є можливість за допомогою оператора **Redim** перевизначити розмірність масиву:

```
Redim [Preserve] varname(subscripts) [As Type] [,  
      varname(subscripts) [As Type]],
```

де *varname* - ім'я існуючого масиву; *subscripts* - розмірність існуючого масиву; *Type* - будь-який тип VBA. Необхідно використовувати окремий оператор *As Type* для кожного масиву, який визначається; *Preserve* - необов'язковий аргумент. Його використання приводить до того, що дані вже присутні в масиві, зберігаються після зміни його розмірності.

Приклади правильного використання оператора **Redim**:

Dim *Array_Month* () **As String** - одномірний строковий динамічний масив.

Redim *Array_Month* (29) – встановлює розмірність динамічного масиву рівну 29 елементам.

Redim *Array_Month* (1 **To** 30) - змінює розмір масиву до 30 елементів.

Redim Preserve *Array_Month* (1 **To** 31) - змінює розмір масиву до 31 елемента, зберігаючи вміст.



Завдання 1

Організувати введення й вивід одномірного масиву, що складається з 10 чисел та двовимірного масиву розміром 3*5.



Аналіз проекту

Введення масиву фіксованої довжини можна здійснити як із клавіатури (використовуючи функцію **Inputbox**), так і з аркуша *Ms Excel*. Вивід масиву також можна здійснити як на екран, за допомогою оператора **Msgbox**, так і на аркуш *Ms Excel*. Для введення найпростіше використовувати цикл **For** із числом кроків, рівним довжині (кількості елементів) масиву.



Порядок виконання

1. Запустимо *MS Excel* і перейдемо в середовище VBA, нажавши клавіші **ALT+F11**.

2. Відкрийте вікно коду VBA, клацнувши на кнопку **View code**  у вікні властивостей **Project** (або виконайте подвійне клацання мишкою на пункті **Thisdocument**).

3. Перейдіть у вікно коду й введіть текст процедури, яка забезпечує введення масиву із клавіатури.

Для одномірного масиву:

```
Option Base 1          'індекси масивів починаються з 1  
Sub Vvod_1 ()
```

```
Dim Mas(10) As Integer 'одномірний масив 10 елем.  
Dim i As Integer  
For i = 1 To 10  
    Mas(i) = Val(Inputbox("Введіть " & i & "-й елемент  
масиву"))  
Next  
End Sub
```

Для двовимірного масиву:

```
Sub Vvod_2()  
    Dim Mas(3,5) As Integer 'двовимірний масив 3 на 5.  
    Dim i As Integer  
    For i = 1 To 3            'цикл по рядках.  
        For j = 1 To 5        'цикл по стовпцях.  
            Mas(i,j) = Val(Inputbox("Введіть елемент масиву. "  
& i & "-й рядок "& j & "-й стовпець."))  
        Next  
    Next  
End Sub
```

4. Розглянемо введення масиву з робочого аркуша Ms Excel. Одномірний масив розташований у діапазоні A1:A10, двовимірний – A1:E3. Нижче наведені відповідні фрагменти коду:

Для одномірного масиву:

```
For i = 1 To 10  
    Mas(i) = Cells(i, 1) 'номер рядка міняється в  
                        'циклі, стовпець перший (A)  
Next
```

Для двовимірного масиву:

```
For i = 1 To 3            'цикл по рядках.  
    For j = 1 To 5        'цикл по стовпцях.  
        Mas(i,j) = Cells(i, j) 'номер рядка й стовпця  
                               'мінється в циклі  
    Next  
Next
```

5. Аналогічно розглянемо вивід масиву на екран і на робочий аркуш Ms Excel, наприклад у діапазон C1:C10. Для цього необхідно замінити функції введення на функції виводу. Фрагменти процедур наведені нижче:

Для одномірного масиву:

```
For i = 1 To 10  
    MsgBox (Str(Mas(i)))  
Next
```

```
For i = 1 To 10  
    Cells(i, 3) = Mas(i) 'вивід масиву в третій  
                        'стовпець (C1:C10)  
Next
```

Для двовимірного масиву:

```
For i = 1 To 3                                'цикл по рядках.  
  For j = 1 To 5                              'цикл по стовпцях.  
    MsgBox (Str(Mas(i, j)))  
  Next  
Next
```

```
For i = 1 To 3  
  For j = 1 To 5  
    Cells(i + 7, j) = Mas(i, j) 'вивід масиву  
                                'на 7 рядків нижче.  
  Next  
Next
```

6. Запустіть на виконання будь-яку процедуру, клацнувши на кнопці **Run**  (або натисніть клавішу **F5**) і перевірте її працездатність.

7. Законспекуйте елементи коду VBA, які призначені для організації введення та виведення масивів.



При організації введення значень масивів з робочого аркуша *MS Excel* слід пам'ятати, що перед запуском процедури необхідно ввести числові значення у відповідні комірки робочого аркуша, інакше масив буде заповнений нульовими значеннями.



Завдання 2

Знайти максимальне значення, а також поміняти місцями перший позитивний і перший негативний елементи масиву *A*, що складається з 10 цілих чисел.



Аналіз проекту

Алгоритм складається з п'яти частин: 1) введення масиву, 2) пошук максимального значення, 3) пошук першого позитивного елементу, 4) пошук першого негативного елементу, 5) обмін їх значень. Для демонстрації результату необхідна ще одна частина - 6) виведення отриманого масиву. Для наочності будемо вводити масив з аркуша *Excel* (наприклад, з діапазону *A1:A10*) і виводити результат на той же аркуш (наприклад, у діапазон *C1:C10*).

Для пошуку максимального значення кожний елемент $A(i)$ масиву порівнюється із уже отриманим на попередніх кроках максимумом. У якості початкового значення можна вибрати будь-який елемент $A(i)$, але зручніше $A(1)$ - циклічну перевірку можна тоді почати із другого елемента, скоротивши число кроків.

Пошук першого позитивного елемента проведемо, перебираючи в циклі значення A . Як тільки зустрінеться $A(k) > 0$, потрібно запам'ятати його індекс (номер k) і відразу вийти із цього циклу. Зверніть увагу, що якщо продовжити цикл, то ми в результаті одержимо номер не першого, а останнього позитивного елемента масиву.

Аналогічно проводимо пошук першого негативного елемента масиву. Нехай це буде $A(j)$. Індекс j запам'ятовуємо.

Тепер залишилося обміняти значення $A(k)$ і $A(j)$, використовуючи змінну - посередника. Після завершення операцій масив виводиться на робочий аркуш *Ms Excel*.



Порядок виконання

1. Запустити Ms Excel, перейти у вікно коду VBA і ввести наступний текст процедури:

```
Option Base 1 'індекси масивів починаються з 1
Sub Obmen()
Dim i, j, k, rab, max As Integer
Dim A(10) As Integer
'вводимо масив з аркуша Excel
For i = 1 To 10
A(i) = Cells(i, 1)
Next
'Знаходимо максимальний елемент масиву
max = A(1)
For i = 2 To 10
If A(i) > max Then max = A(i)
Next
'Знаходимо перший позитивний елемент
For i = 1 To 10
If A(i) > 0 Then
k = i
Exit For
End If
Next
'Знаходимо перший негативний елемент
For i = 1 To 10
If A(i) < 0 Then
j = i
Exit For
End If
Next
'міняємо значення місцями
rab = A(k)
A(k) = A(j)
```

```
A(j) = rab  
'Виведення максимального значення в комірку B(1)  
Cells(1, 2) = "max=" & max  
'Виведення масиву на аркуш Excel у стовпець C  
For i = 1 To 10  
Cells(i, 3) = A(i)  
Next  
End Sub
```

2. На робочому аркуші в діапазон комірок A1:A10 увести вхідні дані для елементів масиву.
3. Запустіть програму на виконання, клацнувши на кнопці **Run**  (або натисніть клавішу **F5**). Проаналізуйте працездатність програми.
4. Збережіть документ у файлі *Лаб7_1.xls* у своїй папці на жорсткому диску.



Завдання 3

Створити форму для сортування одномірних масивів чисел довільної довжини по зростанню.



Аналіз проекту

Сортування – розподіл елементів по групах відповідно до певних правил. Існує досить велика кількість методів сортування. Приведемо два найпростіші з них.

Лінійне сортування. Нехай необхідно впорядкувати масив по зростанню (убуванню) елементів. Алгоритм:

Переглядаємо елементи, починаючи з 1-го. Шукаємо мінімальний (максимальний). Міняємо його місцями з 1-м елементом. Потім переглядаємо елементи, починаючи з 2-го. Шукаємо мінімальний (максимальний). Міняємо його місцями з 2-м елементом. І таким чином до передостаннього елемента.

Сортування методом «бульбашки». Даний метод одержав таку назву за аналогією з бульбашками повітря в склянці води. Більш «легкі» (максимальні або мінімальні) елементи поступово «спливають». На відміну від лінійного сортування, порівнюються тільки пари сусідніх елементів, а не кожний елемент із усіма (тому таке сортування виконується за менше число кроків і, отже, швидше). Алгоритм:

Послідовно розглядаємо пари сусідніх елементів масиву A (тобто порівнюємо перший елемент із другим, другий із третім і так далі). Якщо для сусідніх елементів виконується умова $A(i-1) > A(i)$, то значення міняються місцями. Виконавши один прохід по масиву, одержимо нову послідовність значень. Ця послідовність до кінця не відсортована, але найбільший елемент

перебуває на правій границі масиву. Виконуємо нові проходи, щораз скорочуючи кількість елементів, що переглядаються, на 1.

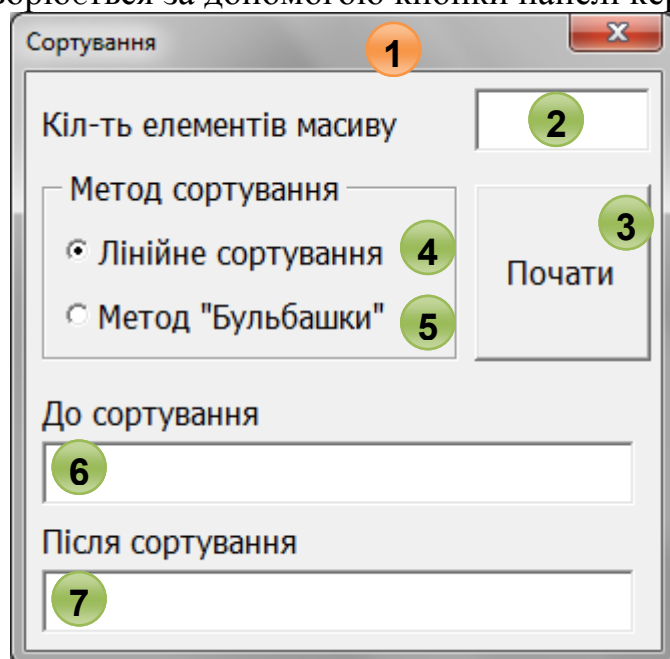


Порядок виконання

1. Перебуваючи в середовищі *Ms Word* або іншого додатку з *Microsoft Office*, натисніть **ALT+ F11**, щоб перейти в середовище VBA.

2. На панелі інструментів клацніть на кнопку **Insert Userform** . З'являється вікно з формою *Userform1*.

3. Створіть форму згідно з ескізом, установивши у вікні властивостей **Properties** значення властивостей (**Name**) і **Caption**, які наведені нижче. Елемент керування для вибору методу сортування має назву **Frame** і створюється за допомогою кнопки панелі керування, яка має вигляд .



Властивість (**Name**)

- 1 — *frmSort*
- 2 — *txtn*
- 3 — *CmdGo*
- 4 — *OptSort1*, **Value=True**
- 5 — *OptSort2*
- 6 — *TxtTo*
- 7 — *TxtAfter*

4. Перейдіть у вікно коду, двічі клацнувши на кнопку **Почати**, і введіть наступний текст процедури:

```
Option Base 1 'індекси масивів починаються з 1
Private Sub CmdGo_Click()
    Dim Mas() As Integer 'динамічний масив
    Dim i, j, n, vr As Integer
    'Введення значень із клавіатури
    n = Cint(Txtn)
    Redim Mas(n) 'задаємо довжину масиву
    For i = 1 To n
        Mas(i) = Val(Inputbox("Введіть " & i & "-й елемент масиву"))
    Next
    'Вивід вихідного масиву
    Txtto = "" 'очищення поля виводу
```



```
For i = 1 To n
    Txtto = Txtto & Mas(i) & "  "
Next
If Optsort1 Then 'Лінійне сортування
For i = 1 To n - 1 'Рухаємося по масиву, скорочуючи
    'невідсортовану частину
For j = i + 1 To n 'Порівнюємо по черзі i-й елемент
    'невідсортованої частини масиву
    'з усіма, від i+ 1-го до кінця
If Mas(j) < Mas(i) Then 'якщо в невідсортованій
    'частині масиву знайшли елемент,
    'більший i-го, то міняємо їх місцями:
vr = Mas(i)
Mas(i) = Mas(j)
Mas(j) = vr
End If
Next
Next
Else 'Метод «бульбашки»
For i = 2 To n
For j = n To i Step -1
If Mas(j - 1) > Mas(j) Then 'витиснути елемент
    'праворуч, тоді бульбашка
    'спливе вліво

vr = Mas(j - 1)
Mas(j - 1) = Mas(j)
Mas(j) = vr
End If
Next
Next
End If
'вивід масиву після сортування
Txtafter = "" 'очищення поля виводу
For i = 1 To n
    Txtafter = Txtafter & Mas(i) & "  "
Next
End Sub
```

5. Запустіть програму на виконання, клацнувши на кнопці **Run**  (або натисніть клавішу **F5**). Проаналізуйте працездатність програми шляхом уведення різних масивів і вибору різних методів сортування.

6. Збережіть документ у файлі *Лаб7_2.doc* у своїй папці на жорсткому диску.



Завдання 4

Дано масив $A(n, m)$. Занести всі негативні елементи масиву в масив B і надрукувати його. Введення й виведення даних здійснювати за допомогою робочого аркуша *Excel*.



Аналіз проекту

Алгоритм завдання складається із трьох частин: ввести матрицю A ; перевірити кожний її елемент: якщо він негативний, помістити його в масив B ; вивести отриманий масив B на друк. Перша й третя частини - введення й виведення даних вирішуються стандартним способом вводу та виводу масивів. Другу частину розглянемо докладніше. Аналіз елементів $A(i, j)$ матриці A можна провести, як звичайно для матриць, за допомогою вкладених циклів **For**, що перераховують усі рядки (i) і стовпці (j) матриці. Якщо $A(i, j)$ - негативний, потрібно привласнити черговому елементу масиву B це значення. Оскільки заздалегідь кількість негативних елементів в A невідомо, то заздалегідь невідома й точна довжина масиву B . Очевидно тільки, що вона не перевищить $n * m$ - кількості елементів вихідної матриці A . Ясно також, що лічильник елементів масиву B ніяк не залежить від чергових i і j . Звідси випливає головний висновок для розв'язку завдань такого типу: для масиву-приймача необхідно завести свій, незалежний лічильник елементів. Відведемо для цього окрему змінну. Нехай це буде k . При записі негативного $A(i, j)$ в $B(k)$ лічильник k буде збільшуватися на одиницю, і по закінченню всієї роботи значення k покаже дійсну довжину масиву B . Для прикладу в програмі взята вихідна матриця A розміром $n=4, m=6$.



Порядок виконання

1. Запустити Ms Excel, перейти у вікно коду VBA і ввести наступний текст процедури:

```
Option Base 1 'індекси масивів починаються з 1
Sub Перенос()
Dim A(4, 6) As Integer
Dim B(24) As Integer, k As Integer
For i = 1 To 4
For j = 1 To 6
A(i, j) = Cells(i, j) 'введення елементів масиву
                        'з діапазону A1:F4

Next
Next
k = 0 'обнуління лічильника масиву B
For i = 1 To 4
```

```
For j = 1 To 6
If A(i, j) < 0 Then
k = k + 1
B(k) = A(i, j)
End If
Next
Next
For i = 1 To k
Cells(9, i) = B(i) 'Виведення масиву B у рядок 9
Next
End Sub
```

2. На робочому аркуші в діапазон комірок A1:F4 ввести вихідні дані для елементів масиву A(4,6).
3. Запустіть програму на виконання, клацнувши на кнопці **Run**  (або натисніть клавішу **F5**). Проаналізуйте працездатність програми.
4. Збережіть документ у файлі *Лаб7_3.xls* у своїй папці на жорсткому диску.



Завдання 5

Існує масив A(8,8) елементів. Підрахувати число парних чисел у матриці й підрахувати суму елементів, що розташовані на парних позиціях рядків і стовпців матриці.



Аналіз проекту

Алгоритм першої частини завдання включає введення матриці, перебір її значень із перевіркою на парність, підрахунок суми парних значень і вивід результату.

Для програмування завдання виберемо змінну (наприклад, k) у якості лічильника парних чисел. Перевірку на парність реалізуємо за допомогою функції **Mod**.

У другій частині вже не важлива парність елементів, важлива парність позицій: елемент A(2,4), наприклад, підходить, а елемент A(3,4) – ні. Щоб потрапити на парну позицію, непотрібно використовувати функцію **Mod** для лічильників рядків i і стовпців j . Досить змінювати крок кожного циклу зі кроком 2.



Порядок виконання

1. Перейти у вікно коду VBA і ввести наступний текст процедури:

```
Option Base 1 'індекси масивів починаються з 1
```

```
Sub matrixnumber()  
Dim i, j, k, s, A(8, 8) As Integer  
For i = 1 To 8 ' вводимо матрицю із клавіатури:  
For j = 1 To 8  
A(i, j) = Val(Inputbox("Введіть елемент масиву. " &  
i & "-й рядок " & j & "-й стовпець."))  
Next  
Next  
k = 0 'обнуління лічильника парних чисел  
'починаємо перебір елементів матриці:  
For i = 1 To 8  
For j = 1 To 8  
'перевіряємо на парність  
If A(i, j) Mod 2 = 0 Then  
k = k + 1 'рахуємо парні значення  
End If  
Next  
Next  
'друкуємо результат  
Msgbox (" Кіл-ть парних чисел: " & k)  
s = 0 'обнуління суми чисел  
For i = 2 To 7 Step 2 'починаємо перебір парних  
For j = 2 To 7 Step 2 'елементів матриці:  
s = s + A(i, j) 'накопичуємо суму  
Next  
Next  
Msgbox ("Сума елементів на парних позиціях: " & s)  
End Sub
```

2. Запустіть програму на виконання, клацнувши на кнопці **Run**  (або натисніть клавішу **F5**). Проаналізуйте працездатність програми.
3. Збережіть документ у файлі *Лаб7_7.xls* у своїй папці на жорсткому диску.



Завдання 6

Існує масив $A(5,5)$ елементів. Підрахувати середнє значення елементів матриці, розташованих строго нижче головної діагоналі матриці.



Аналіз проекту

У завданнях, пов'язаних з діагоналями матриць, слід звертати увагу на залежність значень індексу j (лічильника стовпців) від i (лічильника рядків). Якщо виявлена закономірність, то можна позбутися безлічі зайвих операторів і

зробити програму більш виразною й компактною. Наприклад, усі елементи головної діагоналі мають номери стовпців, що збігаються з номерами рядків, тобто «типовий» представник цієї діагоналі - елемент $A(i,i)$. Причому, для кожного рядка цієї частини матриці, номери стовпців потрібних елементів змінюються від першого до того, що перетинає діагональ. На діагоналі номер стовпця збігається з номером рядка, тобто $j = i$. Оскільки діагональ за умовою не включена, то j змінюється не до i , а до $i - 1$.



Порядок виконання

1. Перейти у вікно коду VBA і ввести наступний текст процедури:

```
Option Base 1 'індекси масивів починаються з 1
Sub Matrix()
Dim A(5, 5) As Integer, S As Integer
Dim k As Integer, Avg As Single
For i = 1 To 5 ' вводимо матрицю із клавіатури:
For j = 1 To 5
A(i, j) = Val(Inputbox("Введіть елемент масиву. " &
i & "-й рядок " & j & "-й стовпець."))
Next
Next
s = 0
k = 0
'номера рядків беруться від 2 до кінця матриці
For i = 2 To 5
'номера стовпців беруться від 1 до діагоналі
For j = 1 To i - 1
S = S + A(i, j)
k = k + 1
Next
Next
Avg = S / k
Msgbox ("Середнє значення елементів= " & Avg)
End Sub
```

2. Запустіть програму на виконання, клацнувши на кнопці **Run**  (або натисніть клавішу **F5**). Проаналізуйте працездатність програми.
3. Збережіть документ у файлі *Лаб7_5.xls* у своїй папці на жорсткому диску.



Завдання для самостійної роботи

1. Знайти мінімальний з елементів масиву $A(10)$, що належить інтервалу $(2; 14)$.
2. Скласти масив B з непозитивних елементів масиву $A(15)$.
3. Порахувати кількість нульових елементів у матриці $A(5 \times 5)$.
4. Перенести елементи, кратні трьом, з масиву $A(5 \times 5)$ у масив B .
5. Обнулити елементи матриці $M(5 \times 5)$, що лежать нижче її побічної діагоналі, включаючи діагональні.



Результат роботи

1. Файл *Лаб7_2.doc* з формою *Сортування*
2. Файли *Лаб7_1.xls*, *Лаб7_3.xls*, *Лаб7_4.xls*, *Лаб7_5.xls* з виконаними завданнями.



Питання для самоконтролю

1. Що таке масив і для чого він створюється? Як організувати доступ до даних у масиві?
2. Що таке нумерація з нульовою базою і як її можна змінити?
3. Що таке одномірний, двомірний і тривимірний масиви. Чому дорівнює гранична кількість вимірів? Чим відрізняються статичний і динамічний масиви?
4. Приведіть синтаксис опису статичного й динамічного масиву в програмі.