Міністерство освіти і науки України
Державний вищий навчальний заклад
“Ужгородський національний університет”
Математичний факультет
Кафедра системного аналізу і теорії оптимізації

ВИКОРИСТАННЯ МАСИВІВ У DELPHI

Методичні вказівки до лабораторних робіт для студентів II-го курсу математичного факультету спеціальності "прикладна математика"

Ужгород – 2011

Семйон І.В., Чупов С.В., Брила А.Ю., Апшай Н.І. Використання масивів у Delphi. Методичні вказівки до лабораторних робіт для студентів ІІ-го курсу математичного факультету спеціальності "прикладна математика". – Ужгород, 2011. – 50с.

Розглядаються одновимірні та багатовимірні масиви як статичні так і динамічні змінні. Використання кожного із типів масивів ілюструється на прикладах, виконаних у системі програмування Delphi.

Рекомендовано до друку Вченою радою математичного факультету

ДВНЗ "Ужгородський національний університет"

від 10 березня 2011 року, протокол № 3.

Поняття масиву

Як відомо, будь-яку задачу людина, як і машина, розв’язує виконуючи кроки деякого алгоритму. Алгоритмом називають наперед задану скінчену послідовність чітких правил або команд (кроків), виконання яких призводить до одержання розв'язку задачі. Програма ж є одним із способів запису алгоритму.

У своїй роботі програміст виконує маніпуляції з величинами. З точки зору програмування, величини – це дані, що обробляються програмами. Носіями даних у програмах є константи, змінні (значення яких зберігається в оперативній пам'яті) та файли (які зберігаються на носіях інформації).

Константи – це величини, значення яких у процесі виконання програми не змінюється. Змінні – це величини, значення яких у процесі виконання програми можуть змінюватися. Імена констант та змінних, як і інших програмних об'єктів, записують у формі ідентифікаторів (послідовності символів латинського алфавіти, цифр та символу нижнього підкреслювання “_”, яка не може починатися з цифри). Кожна змінна і константа належать до визначеного типу. Тип даних – це сукупність властивостей певного набору даних, від яких залежать діапазон значень, якого можуть набувати ці дані, а також сукупність операцій, які можна виконувати над цими даними.

Усі типи даних розділяють на дві групи: скалярні (прості) та структуровані (складені). До скалярних відносять типи даних (цілі, дійсні, символьний, рядковий, логічний), які дозволяють опрацьовувати окремі числа, символи, рядки і т.д. Але у більшості задач доводиться розглядати сукупності величин одного типу (вектори, матриці, таблиці, числові послідовності і т.д.). Звичайно, можна було би для кожної величини ввести змінну. Але уявіть собі, що буде у випадку, коли кількість величин рівна 100. Тому для збереження сукупності елементів одного типу у мовах програмування визначається так званий структурований тип даних – *масив*.

Можна дати кілька означень масиву.

Масив – це структура даних, що являє собою однорідну (за типом), фіксовану (за розміром і конфігурацією) сукупність нумерованих елементів.

Масивом називається скінчена послідовність елементів одного типу, які різняться порядковим номером.

Масив – формальне об'єднання декількох однотипних об'єктів (чисел, символів, рядків і т.д.), які розглядаються як єдине ціле

Масиви відносяться до структур з прямим або довільним доступом. Щоб визначити окремий елемент масиву потрібно вказати його *індекс*. В одновимірному масиві індекс – це порядковий номер елементу масиву. В якості індексів можуть використовуватися значення простих порядкових типів (цілі, символьний, злічені). Кількість індексів таблиці називають *розмірністю*, кількість дозволених значень кожного індексу – *діапазоном*, а сукупність розмірності та діапазону – *формою масиву.*

Класифікація масивів може здійснюватися за багатьма ознаками:

1) можливістю зміни значень елементів масиву (масиви-константи та масиви-змінні);

2) кількістю індексів, які необхідно вказати для того, щоб ідентифікувати елемент у масиві (масиви одновимірні, двовимірні, тривимірні і т.д.);

3) можливістю зміни кількості елементів масиву в процесі виконання програми (масиви фіксованого розміру та змінного розміру);

4) можливістю зміни типу елементів масиву в процесі виконання програми (масиви з фіксованим типом елементів та масиви із змінним типом елементів).

1. Одновимірні масиви

1.1. Статичні одновимірні масиви фіксованого розміру та фіксованого типу елементів

Одновимірний масив часто асоціюють із вектором. Загальний вигляд опису типу одновимірного масиву:

```
TYPE
<ім'я типу масиву>=array[<поч.зн.індексу>..<кін.зн.індексу>] of <тип елементів>;
```

Ім'я типу масиву задається у формі ідентифікатора. Як завжди, ім'я типу прийнято починати з великої літери "T" (TVector, TYears,TStudents та ін.). Початкове та кінцеве значення індексу при заданні діапазона зміни індексу повинні бути константами порядкового типу (цілого, символьного, зліченого).

Приклад.

```
CONST n=10;
TYPE  TVectA = array[1..10] of real;
      TVectB = array[0..30] of byte;
      TVectA1= array[1..n] of string;
      TVectA2= array['a'..'f'] of real;
VAR   A: TVectA;
      B: TVectB;
      A1: TVectA1;
      A2: TVectA2;
```

Для ідентифікації елемента масиву необхідно вказати ім'я масиву та індекс елемента, який записується у квадратних дужках після імені масиву

```
<ім'я масиву> [ <індекс> ]
```

Приклад.

```
A [ 1 ] := 2;
A [ 4 ] := B[1]+2.4;
A1 [ 3 ] := ' Мир ' ;
A2 [ 'b' ] := 55;
```

Наведемо декілька важливих зауважень:

1) елементи масиву можуть бути довільного, але тільки одного типу;

2) межі зміни індексів можуть бути тільки сталими величинами;

3) діапазон не може змінюватися під час виконання програми;

4) значення індексу не повинно виходити за межі описаного діапазону;

5) оскільки процедури введення/виведення розраховані на введення та виведення значень простих типів, то масив потрібно вводити/виводити поелементно, тобто кожен елемент окремо.

Як було зауважено, межі зміни діапазону індексів повинні бути константами, і діапазон не може змінюватися під час виконання програми. Але ж при розв'язанні задач, як правило, кількість елементів масиву вводить користувач під час виконання програми. Тому в залежності від задачі при описі масиву необхідно вказати гранично можливий діапазон зміни індексу. Для збереження ж справжнього діапазону зміни індексу описують додаткові змінні, значення яких не можуть виходити за вказані гранично можливі.

Приклад. Якщо необхідно зберігати роки народження учнів у класі (учнів не більше 40), то тип масив описують так.

```
TYPE
    TVectYears = array[1..40] of integer;
```

Для збереження дійсної кількості учнів у класі описуємо додатково змінну цілого типу n, значення якої буде задавати користувач під час виконання програми. Зрозуміло, що значення змінної n у даному випадку не може бути більшим за 40.

```
VAR
    Y : TVectYears;        { Y – масив років народження }
    n : integer;           { n – кількість учнів у класі (n ≤ 40)}
```

Тут

Y[1] – рік народження 1-го учня;

Y[2] – рік народження 2-го учня;

……

Y[n] – рік народження n-го учня.

Наведемо ще декілька прикладів опису одновимірних масивів для розв'язання відповідних задач

Приклад. Необхідно знайти загальну масу автомобілів у парку.

Будемо вважати, що кількість автомобілів не перевищує 100.

```
TYPE
     TVectCars = array[1..100] of Real;
VAR
     C : TVectCars;          { C – масив для збереження
                                маси а втомобілів }
     n : integer;            { n – кількість автомобілів ( n ≤ 100 )}
```

Тут

C[1] – маса 1-го автомобіля;

C[2] – маса 2-го автомобіля;

……

C[n] – маса n-го автомобіля.

Приклад. Необхідно зберігати кількість маленьких букв у слові на англійській мові.

У даному випадку немає потреби вводити додаткову змінну для збереження кількості елементів масиву, оскільки кожна із букв алфавіту може входити у слово.

```
TYPE
     TVectChars = array['a' .. 'z'] of byte;
VAR
     C : TVectChars;         { C – масив для збереження
                               кількості входжень букв у слово }
```

Тут

C['a'] – кількість входжень букви 'a' ;

C['b'] – кількість входжень букви 'b';

……

C['z'] – кількість входжень букви 'z'.

Представлення одновимірного масиву в пам'яті ЕОМ

Для збереження масивів в пам'яті ЕОМ використовується послідовне представлення. Тобто для збереження масиву наперед виділяється ділянка оперативної пам'яті величиною

<кількість елементів> * <кількість байтів для збереження одного елемента>

і елементи розміщуються послідовно один за одним.

Приклад.

```
TYPE   TVectA = array[1..10] of real;
VAR    A: TVectA;
```

Змінна A в пам’яті ЕОМ буде розміщена наступним чином

							Комірки
…	A[1]	A[2]	A[3]	…	A[10]	…	**Імена комірок**
	Область масиву A						

Приклад.

```
TYPE   TVectB = array[3..8] of real;
VAR    B: TVectB;
```

Змінна B в пам’яті ЕОМ буде розміщена наступним чином

								Комірки
…	B[3]	B[4]	B[5]	B[6]	B[7]	B[8]	…	**Імена комірок**
	Область масиву B							

Приклад.

```
TYPE   TVectC = array['a'..'e'] of real;
VAR    C: TVectC;
```

Змінна A в пам’яті ЕОМ буде розміщена наступним чином

							Комірки
…	C['a']	C['b']	C['c']	C['d']	C['e']	…	**Імена комірок**
	Область масиву C						

Введення елементів масиву

Масив відноситься до структурованих типів даних, тому введення і виведення масивів повинно здійснюватися поелементно. Як правило, введення елементів масиву здійснюється у циклі. Загальний вигляд циклу введення елементів масиву може виглядати наступним чи ном:

```
for <параметр> := <поч.зн.інд.> to <кін. зн.інд.> do
begin
        write('<ім"я мас.>[ ', <параметр> ,' ]= ');
        readln(<ім"я мас.> [ <параметр>  ] );
end;
```

Приклад.

```
for  i := 1  to  n  do
begin
        write(' Y [ ', i , ' ] = ');
        readln( Y [ i ] );
end;
```

Приклад.

```
for  i := 1  to  n  do
begin
      write(‘ Введіть масу  ’, i , ’ - го автомобіля : ’);
      readln( C [ i ] );
end;
```

Приклад.

```
for  k := ‘a’  to  ‘e’  do
begin
      write(‘ C [ ’, k , ’ ] = ’);
      readln( C [ k ] );
end;
```

Виведення елементів масиву

Як і введення, виведення елементів масиву також здійснюється поелементно. Загальний вигляд циклу виведення елементів масиву може виглядати наступним чином:

```
for <параметр> := <поч.зн.інд.> to <кін. зн.інд.> do
begin
  write(‘<ім”я мас.>[ ’,<параметр>,’ ]= ’,<ім”я мас.>[ <параметр>  ] );
end;
```

Приклад.

```
for  i := 1  to  n  do
begin
      write(‘ Y [ ’, i , ’ ] = ’, Y [ i ] );
end;
```

Приклад.

```
for  k :=  ‘a’  to  ‘e’  do
begin
      write(‘ Кількість букв  «’, k , ’»  у слові =  ’, C [ k ] );
end;
```

Наведемо декілька прикладів задач, розв’язаних з використанням масивів.

Приклад. Дано $a, b \in R^n$, знайти $c = a + b$.

Розв’язання

Для збереження векторів використаємо однойменні одновимірні масиви A, B і C. Будемо вважати, що $n \le 50$.

```
PROGRAM Sum;
{$APPTYPE CONSOLE}
     Type TVector = array[1..50] of real;
     VAR A,B,C: TVector;
```

```
          n,i:integer;
BEGIN
write('Введіть розмірність n ='); readln(n);
{------------  Введення масиву A --------}
     writeln('Введіть елементи масиву A');
     for i:=1 to n do
       begin
         write('A[',i,']= '); readln(A[i]);
       end;
{------------  Введення масиву B --------}
     writeln('Введіть елементи масиву B');
     for i:=1 to n do
       begin
         write('B[',i,']= '); readln(B[i]);
       end;
{============= Знаходження вектора C =============}
     for i:=1 to n do
         C[i]:= A[i]+B[i];
{--------------- Виведення масиву C ----------}
     for i:=1 to n do
        Writeln('  C[',i,']= ',C[i]:6:2);
     readln;
END.
```

Результати роботи програми

```
Введіть розмірність n =3
Введіть елементи масиву A
A[1]= 2
A[2]= 1
A[3]= 5
Введіть елементи масиву B
B[1]= 1
B[2]= 1
B[3]= 0
Елементи масиву C
 C[1]=  3.00
 C[2]=  2.00
 C[3]=  5.00
```

Приклад. Вивести на екран масу тих автомобілів, які важчі за M.

Розв'язання

Для збереження маси автомобілів використаємо масив C. Будемо вважати, що кількість автомобілів $k \le 100$.

```
PROGRAM Sum;
{$APPTYPE CONSOLE}
     Type TCars = array[1..100] of Real;
```

```
        VAR C:TCars;
            k,i:integer;
            M:Real;
    BEGIN
    write('Введіть кількість автомобілів : '); readln(k);
    {------------  Введення маси автомобілів --------}
    writeln('Введіть маси автомобілів');
    for i:=1 to k do
      begin
        write('Маса ',i,'- го автомобіля : ');
         readln(C[i]);
      end;
    write('Введіть M : '); readln(M);
    {-------  Виведення автомобілів важчих за M -----}
    writeln('Автомобілі важчі за ',M:6:2,' : ');
    for i:=1 to k do
      begin
        if C[i]>M then
        writeln(i,'–ий автомобіль з масою ',C[i]:6:2);
      end;
     readln;
    END.
```

Результати роботи програми

```
Введіть кількість автомобілів : 5
Введіть маси автомобілів
Маса 1- го автомобіля : 1000
Маса 2- го автомобіля : 1500
Маса 3- го автомобіля : 800
Маса 4- го автомобіля : 2500
Маса 5- го автомобіля : 1250
Введіть M : 1100
Автомобілі важчі за 1100.00 :
2-ий автомобіль з масою 1500.00
4-ий автомобіль з масою 2500.00
5-ий автомобіль з масою 1250.00
```

Копіювання статичних масивів

Для копіювання значень з одного масиву в інший масив того ж типу можна використати оператор присвоювання

<Масив 1> := < Масив 2> ;

Приклад. Нехай у програмі описано два одновимірних статичних масиви

```
        Type TVector = array[1..3] of real;
        VAR A,B: TVector;
```

Це описання призводить до виділення пам’яті для масивів A і B.

…							…	
…	A[1]	A[2]	A[3]	B[1]	B[2]	B[3]	…	
…	Область масиву A			Область масиву B			…	

Після операторів присвоєння

```
A[1]:=5;    A[2]:=47;    A[3]:=8;
```

одержимо:

…	5	47	8				…	
…	A[1]	A[2]	A[3]	B[1]	B[2]	B[3]	…	
…	Область масиву A			Область масиву B			…	

А виконавши оператор присвоювання

```
B:=A;
```

одержимо:

…	5	47	8	5	47	8	…	
…	A[1]	A[2]	A[3]	B[1]	B[2]	B[3]	…	
…	Область масиву A			Область масиву B			…	

Одновимірні масиви-константи

У мові Паскаль є можливість описання типізованих констант -масивів. При цьому необхідно вказати всі елементи описуваного масиву.

```
CONST
    <ім’я масиву> : array [<поч. зн. інд> .. <кін. зн. інд.>] of <тип елементів> =
(<елемент 1>,<елемент 2 >,…,<елемент N >);
```

Приклад.

```
const Digits:array[‘0’..’3’] of char=(‘0’,’1’,’2’,’3’);
      Digits1:array[‘0’..’3’] of char=(‘0123’);
      A:array[1..4] of real =(1.5, 1.2, 0.6, 7.8);
```

1.2. Динамічні одновимірні масиви з фіксованим типом елементів

Як було зазначено раніше, одновимірні масиви використовують для збереження сукупності елементів одного типу. Але, як правило, визначити необхідну кількість елементів масиву наперед неможливо. Тому виділення пам’яті для максимальної гранично можливої кількості елементів може

призвести до неефективного використання пам'яті. До того ж кількість елементів масиву не може бути збільшеною під час виконання програми. У зв'язку з цим, на практиці використовують динамічні масиви фіксованого типу. Особливістю цих масивів є те, що кількість елементів масиву може динамічно змінюватися під час виконання програми.

При описі динамічних масивів фіксованого розміру кількість елементів масиву не зазначається, а вказується тільки тип його елементів. Наведемо загальний вигляд їх опису:

```
TYPE
    <ім'я типу масиву> = array of <тип елементів>;
```

Приклад.

```
TYPE   TVectA = array of byte;
       TVectB = array of real;
       TVectA1= array of string;
VAR    A: TVectA;
       B: TVectB;
       A1: TVectA1;
```

Змінні типу динамічний масив є вказівниками. Пам'ять для елементів масиву виділяється під час виконання програми з використанням процедури SetLength, загальна форма виклику якої наступна:

SetLength(<ім'я динамічного масиву>,<кількість елементів>);

При цьому буде виділено пам'ять для збереження вказаної кількості елементів (адреса початку пам'яті зберігається у вказівнику – імені динамічного масиву). Нумерація елементів масиву починається від 0.

Приклад.

```
SetLength( A , 10);     //Нумерація від 0 до 9.
SetLength( B , 30);     //Нумерація від 0 до 29.
SetLength( A1 , 25);    //Нумерація від 0 до 24.
```

Індекс першого елемента можна визначати за допомогою функції Low, загальна форма виклику якої наступна:

Low (<ім'я динамічного масиву>);

Індекс останнього елемента можна визначати за допомогою функції High, загальна форма виклику якої наступна:

High (<ім'я динамічного масиву>);

Приклад.

```
TYPE   TVectA = array of Byte;
VAR    A: TVectA;
       First,Last:Integer;
BEGIN
SetLength( A , 10);    //Виділено пам'ять для 10 елементів типу Byte.
First:= Low( A );          // First = 0.
Last:= High( A );          // Last = 9.
End.
```

Не дивлячись на те, що змінна динамічного масиву є вказівником, звертання до елементів масиву можна здійснювати як і при використанні статичних масивів:

```
<ім'я масиву>  [ <індекс> ]
```

Приклад.

```
A [ 0 ] := 2;
B [ 4 ] := B[1]+2.4;
A1 [ 3 ] := ' Мир ' ;
```

Проілюструємо використання динамічних одновимірних масивів на попередньому прикладі.

Приклад. Дано $a,b \in R^n$, знайти $c = a + b$.

Розв'язання

Для збереження векторів використаємо однойменні одновимірні динмічні масиви фіксованого типу A, B і C.

```
PROGRAM Sum;
{$APPTYPE CONSOLE}
      Type TVector = array of real;
      VAR A,B,C: TVector;
            n,i:integer;
BEGIN
write('Введіть розмірність n ='); readln(n);
{-----------  Введення масиву A --------}
  SetLength(A,n+1);          //Для зручності створюємо (n+1) елемент,
                             //щоб номер останнього був n
      writeln('Введіть елементи масиву A');
      for i:=1 to n do
        begin
            write('A[',i,']= '); readln(A[i]);
```

```
          end;
{------------  Введення масиву B --------}
        writeln('Введіть елементи масиву B');
  SetLength(B,n+1);
        for i:=1 to n do
          begin
                write('B[',i,']= '); readln(B[i]);
          end;
{============= Знаходження вектора C ============}
  SetLength(C,n+1);
        for i:=1 to n do
                C[i]:= A[i]+B[i];
{--------------- Виведення масиву C ----------}
        for i:=1 to n do
          Writeln('  C[',i,']= ',C[i]:6:2);
        readln;
END.
```

Результати роботи програми аналогічні до попереднього прикладу.

Представлення динамічного одновимірного масиву фіксованого типу в пам'яті ЕОМ

Як було зазначено, змінна динамічний масив є вказівником, тобто змінною, яка може зберігати адресу оперативної пам'яті ЕОМ. Тому опис

```
TYPE   TVectA = array of Byte;
VAR    A: TVectA;
```

призводить до виділення пам'яті тільки для вказівника A. Нехай змінну A розташовано в оперативній пам'яті за адресою 100.

…	100	…	550	551	552	……	559	…	Адреса
…	nil	…				……		…	Комірки
…	A	…	Невикористана пам'ять					…	**Імена комірок**

Процедура SetLength виділяє пам'ять для збереження необхідної кількості елементів вказаного типу і поміщає адресу початку виділеної пам'яті у змінну-вказівник – ім'я динамічного масиву під час роботи програми. Нехай невикористана ділянка пам'яті починається з адреси 550. Тоді виклик процедури

```
SetLength( A , 10);
```

призводить до виділення пам'яті для десяти елементів типу Byte, з нумерацією елементів від 0 до 9, а адреса початку 550 заноситься у A.

…	100	…	550	551	552	……	559	…	Адреса
…	**550**	…				……		…	Комірки
	A		A[0]	A[1]	A[2]	……	A[9]	…	**Імена комірок**
			Область масиву A						

Копіювання масивів. Функція Copy

Динамічні масиви є вказівниками, тому використання оператора присвоювання не дозволяє здійснити копіювання значень масиву.

Приклад. Нехай у програмі описано два динамічних масиви

```
TYPE   TVect = array of byte;
VAR    A,B: TVect;
```

Такий опис призведе до виділення пам'яті для вказівників A і B. Нехай пам'ять для вказівників виділена починаючи з адреси 100, а невикористана пам'ять починається з адреси 550.

…	100	104	…	550	551	552	……	559	…	Адреса
…	nil	nil	…				……		…	Комірки
…	A	B	…	Невикористана пам'ять					…	**Імена комірок**

Виконавши оператори

```
SetLength( A , 3);    //Виділено пам'ять для 3 елементів типу Byte.
A [ 0 ] := 2;
A [ 1 ] := 7;
A [ 2 ] := 6;
```

одержимо:

…	100	104	…	550	551	552	553	554	555	…	Адреса
…	550	nil	…	2	7	6				…	Комірки
…	A	B	…	A[0]	A[1]	A[2]	Невикористана пам'ять				**Імена комірок**
				Область масиву A							

В результаті виконання оператора присвоювання

```
B:=A;
```

одержимо випадок, коли масиви A і B використовують одну і ту ж ділянку пам'яті:

…	100	104	…	550	551	552	553	554	555	…	Адреса
…	550	550	…	2	7	6				…	Комірки
…	A	B	…	A[0]	A[1]	A[2]	Невикористана пам’ять				**Імена комірок**
				B[0]	B[1]	B[2]					
				Область масивів A і B одночасно							

Тому зміна значень масиву B призведе до зміни значень масиву A. Наприклад, в результаті присвоєння

```
B[1]:=25;
```

одержимо, що значення A[1] також буде рівним 25:

…	100	104	…	550	551	552	553	554	555	…	Адреса
…	550	550	…	2	25	6				…	Комірки
…	A	B	…	A[0]	A[1]	A[2]	Невикористана пам’ять				**Імена комірок**
				B[0]	B[1]	B[2]					
				Область масивів A і B одночасно							

Для коректного копіювання значень масиву A у масив B можна скористатись функцією Copy, загальний формат виклику якої наступний

```
<Динам. масив 1>:=Copy(<Динам. масив 2>);
```

Так, наприклад, якщо вже створено динамічний масив A

…	100	104	…	550	551	552	553	554	555	…	Адреса
…	550	nil	…	2	7	6				…	Комірки
…	A	B	…	A[0]	A[1]	A[2]	Невикористана пам’ять				**Імена комірок**
				Область масиву A							

то виклик функції Copy

```
B:=Copy(A);
```

дає змогу виділити пам’ять для масиву B і скопіювати значення з масиву A.

…	100	104	…	550	551	552	553	554	555	…	Адреса
…	550	553	…	2	7	6	2	7	6	…	Комірки
…	A	B	…	A[0]	A[1]	A[2]	B[0]	B[1]	B[2]	Невикористана пам’ять	**Імена комірок**
				Область масиву A			**Область масиву B**				

Проілюструємо приклад використання функції Copy на прикладі.

Приклад. Дано послідовність з n дійсних чисел $a_1, a_2, ..., a_n$. На основі цієї послідовності сформувати послідовність $c_1, c_2, ..., c_n$, де

$$c_i = a_i + b_i,\ i = 1,2,...,n,$$

а послідовність $b_1, b_2, ..., b_n$ складається з елементів послідовності $a_1, a_2, ..., a_n$, які упорядковані за зростанням.

Розв'язання

```
PROGRAM P_1_1C;
{$APPTYPE CONSOLE}
    Type TVector = array of real;
    VAR  A,B,C: TVector;
         n,i,j,l:integer;
      r:Real;
BEGIN
write('Введіть розмірність n ='); readln(n);
{------------  Введення масиву A --------}
   SetLength(A,n+1);
    writeln('Введіть елементи масиву A');
    for i:=1 to n do
      begin
        write('A[',i,']= '); readln(A[i]);
      end;
{------------  Копіювання масиву A --------}
  B:=Copy(A);
{---     Формування масиву B   ---}
{--- (сортування масиву з елементів масиву A) ----}
    for i:=1 to n-1 do
      begin
       l:=i;
     for j :=i+1  to n do
     begin
      if B[j]<B[l] then
       l:=j;
     end;
     if l<>i then
      begin
       r:=B[l];
       B[l]:=B[i];
       B[i]:=r;
      end;
      end;
{============ Знаходження вектора C ============}
```

```
   SetLength(C,n+1);
    for i:=1 to n do
        C[i]:= A[i]+B[i];
{--------------- Виведення масиву C ----------}
 writeln('Елементи масиву C');
    for i:=1 to n do
       Writeln('  C[',i,']= ',C[i]:6:2);
    readln;
END.
```

Результати роботи програми.

```
Введіть розмірність n = 3
Введіть елементи масиву A
A[1]= 2
A[2]= 1
A[3]= 4
Елементи масиву C
  C[1]=   3.00
  C[2]=   3.00
  C[3]=   8.00
```

1.3. Динамічні одновимірні масиви змінного типу елементів . Variant-масиви

Звичайно, використання динамічних масивів з фіксованим типом елементів дозволяє уникнути неефективного використання пам’яті, оскільки створюється стільки елементів, скільки необхідно. Але на практиці також зустрічаються випадки, коли наперед важко визначити тип елементів, які необхідно зберігати та опрацьовувати.

Наприклад. З клавіатури у довільному порядку вводяться роки народження учнів або їх імена. Необхідно впорядкувати їх за зростанням та вивести на екран три останні. Якщо вводилися роки народження, то вивести їх збільшеними на 5.

Очевидно, в даному випадку використання раніше розглянутих типів масивів не дає можливості ефективно розв’язати поставлену задачу, оскільки наперед невідомо, чи будемо вводити роки народження, чи імена учнів. Саме в таких випадках використовують динамічні одновимірні масиви змінного

типу елементів. Основою таких масивів є тип Variant, тому їх і називають Variant-масивами. Схема їх використання складається з наступних кроків:

1) підключаємо модуль Variants;

Uses Variants;

2) описуємо змінну-масив, як змінну типу Variant;

Var <ім'я масиву> : Variant;

3) виділяємо пам'ять для масиву (при цьому вказуємо початкове, кінцеве значення індексу та тип елементів масиву).

<ім'я масиву>:=VarArrayCreate([<поч.зн. інд.>,<кін.зн.інд>],<тип елем.>);

Зауважимо, індекс може бути тільки цілого типу, а в якості типу елемента масиву може використовуватись один із типів: varEmpty, varNull, varSmallint, varInteger, varSingle, varDouble, varCurrency, varDate, varOleStr, varDispatch, varError, varBoolean, varVariant, varUnknown, varShortInt, varByte, varWord, varLongWord, varInt64, varStrArg, varAny, varTypeMask.

Приклад.

```
    Uses Variants;
    VAR A,B,V: Variant;
        n:integer;
    . . . . . .
BEGIN
    . . . . . .
  A:=VarArrayCreate([0,n],varInteger);
  B:=VarArrayCreate([2,7],varDouble);
  V:=VarArrayCreate([1,n], varVariant);
    . . . . . .
END.
```

Доступ до елементів масиву здійснюється аналогічно до раніше розглянутих типів масивів.

Ще один спосіб створення Variant-масивів полягає у використанні функції VarArrayOf

<ім'я масиву>:=VarArrayOf([<елемент 1>,<елемент 2 >,…,<елемент N >]);

Дана функція виділяє пам'ять для вказаної кількості елементів та заповнює його цими елементами.

Приклад.

```
    Uses Variants;
    VAR A: Variant;
    . . . . . .
BEGIN
    . . . . . .
  A:= VarArrayOf([25, 7, 35, 7]);
                  {Створюється та заповнюється масив з 4 елементів}
    . . . . . .
END.
```

A[4] := VarArrayOf([1, 10, 100, 1000]);

Проілюструємо використання Variant-масивів на прикладі.

Приклад. З клавіатури у довільному порядку вводяться роки народження учнів або їх імена. Необхідно впорядкувати їх за зростанням та вивести на екран три останні. Якщо вводилися роки народження, то вивести їх збільшеними на 5.

```
PROGRAM P_1_2D;
{$APPTYPE CONSOLE}
    Uses Variants;
    VAR A: Variant;
        n,i,j,l,num:integer;
       s:string;
BEGIN
  write('Введіть кількість елементів n ='); readln(n);
 writeln('Виберіть тип елементів:');
 writeln('1. Роки народження.');
 writeln('2. Імена.');
 write('Ваш вибір: ');
 readln(num);
case num of
  1:A:=VarArrayCreate([0,n],varInteger);
  2:A:=VarArrayCreate([0,n],varOleStr);
 end;
{------------  Введення масиву  --------}
    writeln('Введіть елементи ');
    for i:=1 to n do
      begin
       write(i,'-ий елемент : ');
       readln(s);
       A[i]:=s;
      end;
```

```
{------------  Сортування масиву --------}
    for i:=1 to n-1 do
      begin
       l:=i;
     for j :=i+1  to n do
     begin
      if A[j]<A[l] then
       l:=j;
     end;
     if l<>i then
      begin
       A[0]:=A[l];
       A[l]:=A[i];
       A[i]:=A[0];
      end;
      end;
{============ Виведення останніх 3 =============}
  writeln('Останні 3 елементи: ');
  for i:=n-2 to n do
   if num=1 then
    writeln(A[i]+5)
   else
    writeln(A[i]);
    readln;
END.
```

Результати роботи програми

1. Вводяться роки народження

```
Введіть кількість елементів n =5
Виберіть тип елементів:
1. Роки народження.
2. Імена.
Ваш вибір: 1
Введіть елементи
1-ий елемент :  1990
2-ий елемент : 1989
3-ий елемент : 1993
4-ий елемент : 1978
5-ий елемент : 1977
Останні 3 елементи:
1994
1995
1998
```

2. Вводяться імена

```
Введіть кількість елементів n =5
 Виберіть тип елементів:
 1. Роки народження.
 2. Імена.
 Ваш вибір: 2
 Введіть елементи
 1-ий елемент : Stas
 2-ий елемент : Bika
 3-ий елемент : Dora
 4-ий елемент : Andy
 5-ий елемент : Nona
 Останні 3 елементи:
 Dora
 Nona
 Stas
```

Зауваження. Очевидно, Variant-масиви є найбільшим гнучкими, можна задавати як кількість елементів, так і тип елементів. Але слід пам’ятати, що кожен елемент такого масиву типу Variant і займає 16 байтів дорогоцінної оперативної пам’яті. Тому на практиці їх використовують рідко.

Питання для самоконтролю

1. *Що таке одновимірний масив?*
2. *Як задається діапазон зміни індексів?*
3. *Як зберігаються одновимірні масиви в пам'яті ЕОМ?*
4. *Які особливості введення одновимірного ма сиву?*
5. *Які особливості виведення одновимірного масиву?*
6. *Як описуються одновимірні масиви-константи?*
7. *Як описуються динамічні масиви фіксованого типу?*
8. *Чи відрізняються статичні масиви від динамічних масивів фіксованого типу?*
9. *Яка особливість змінної динамічног о масиву фіксованого типу?*
10. *Які межі зміни індексу у динамічних масивах фіксованого типу?*
11. *Яка процедура використовуються для виділення пам'яті для динамічних масивів фіксованого типу?*
12. *Як визначити максимальне значення індексу динамічного масиву фіксованого типу?*
13. *Як здійснюється копіювання значень динамічного масиву фіксованого типу?*
14. *Що таке динамічний масив змінного типу?*
15. *Як виділяється пам'ять для динамічних масивів змінного типу?*
16. *Якого типу є змінна-динамічний масив змінного типу?*

Завдання для самостійної роботи

1. Дано n дійсних чисел: $x_1, x_2, ..., x_n$. Знайти середнє арифметичне та середнє геометричне значення цих чисел.
2. Дано n дійсних чисел: $x_1, x_2, ..., x_n$. Знайти найбільше та найменше серед них.
3. Дано n дійсних чисел: $x_1, x_2, ..., x_n$. Знайти найбільше серед від'ємних та найменше серед додатних.
4. Дано одновимірний масив, який містить послідовність дійсних чисел. З'ясувати, скільки серед елементів цієї послідовності є пар з трьох елементів, які слідують підряд і утворюють арифметичну прогресію.
5. Знайти добуток від'ємних елементів вектора $a \in R^n$.
6. Знайти кількість від'ємних елементів у векторі $x \in R^n$, які розташовані після першого додатного.
7. Знайти найбільший елемент серед елементів вектора $x \in R^n$ з парними індексами.
8. У векторі $x \in R^n$ всі елементи, які більші за середнє арифметичне замінити нулями.
9. Дано: $n \in N$, $x, y \in R^n$. Побудувати вектор z, який містить спочатку додатні координати вектора x а потім додатні координати вектора y.
10. Дано одновимірний масив дійсних чисел. Знайти суму елементів, які розташовані до першого від'ємного елемента.
11. Дано одновимірний масив дійсних чисел X. В цьому масиві поміняти місцями елементи, що розташовані симетрично відносно середини.
12. Дано одновимірний масив, який містить послідовність дійсних чисел. Для кожної з пар з трьох елементів (з номерами : 1,2,3, 4, 5, 6, 7, 8, 9 …) найбільший елемент замінити середнім арифметичним.
13. Дано : $x \in R^n$. Знайти $y = (x_1, x_1 + x_2, ..., x_1 + x_2 + ... + x_n)$.
14. Дано : $x, y \in R^n$. Знайти $z = (x_1, y_1, x_2, y_2, ..., x_n, y_n)$.

15. Дано одновимірний масив дійсних чисел. З'ясувати, чи утворюють елементи масиву арифметичну прогресію.
16. Дано одновимірний масив дійсних чисел. З'ясувати, чи утворюють елементи масиву геометричну прогресію.
17. Дано одновимірний масив дійсних чисел X. Знайти найменший елемент серед додатних елементів цього масиву.
18. Дано одновимірний масив дійсних чисел X. Знайти найбільший елемент серед від'ємних елементів цього масиву.
19. Дано послідовність із n дійсних чисел. Визначити скільки з них більше своїх "су сідів" (попереднього і наступного чисел).
20. Дано послідовність із n дійсних чисел. Знайти порядковий номер того з них, дробова частина якого є найменшою.

21. Дано послідовність ненульових цілих чисел. Визначити скільки разів у цій по - слідовності змінюється на протилежний знак її елементів.

22. Дано два вектори $\bar{a}(a_1, a_2, ..., a_n)$ і $\bar{b}(b_1, b_2, ..., b_n)$. Підрахуйте кількість тих координат векторів $\bar{a}$ і $\bar{b}$, для яких $a_i < b_i$.
23. Побудувати масив $A=(a_i)$, елементи якого задаються формулою:

$$a_i = \frac{1+2+\cdots+i}{(1-2+3-4+\cdots+(-1)^{i+1}i)}, \quad (i=1,2,...,n).$$

24. Обчислити добуток від'ємних елементів масиву *A*.
25. Побудувати масив *A*=(a_i), елементи якого задаються формулою:

$$a_i = \sin x + 2\sin(1+2)x + \cdots + i\sin(1+2+\cdots+i)x, \ (i=1,2,...,n).$$

26. Знайти найбільший елемент масиву *A*.
27. Задано масиви $A=(a_i)$ та $B=(b_i)$ $(i=1,2,...,n)$. Побудувати масив C, в якому на парних місцях послідовно розташовані елементи масиву *A*, а на непарних *B*.

28. Побудувати масив *A*=(a_i), елементи якого задаються формулою:

$$a_1 = a_2 = x, \quad a_3 = y, \ a_i = a_{i-2} + \frac{a_{i-1}}{2^{i-1}}a_{i-3}, \quad (i=4,5,...,n),$$

29. де *x, y* – вводяться з клавіатури. Вивести кількість елементів масиву *A*, які більші за задане число *z*.

30. Задано масив $B=(b_i)$, $(i=1,2,...,n)$ де $b_i = \begin{cases} 1+\frac{1}{2}+\ldots+\frac{1}{i}, & \textit{якщо } i-\textit{парне}, \\ i!/2+3, & \textit{якщо } i-\textit{непарне}. \end{cases}$

31. Знайти добуток елементів масиву *B* з непарними номерами.

32. Побудувати масив *A*=(a_i), елементи якого задаються формулою:

$$a_i = \frac{-1+2-3+\cdots+(-1)^i i}{i!}, \quad (i=1,2,...,n).$$

33. Обчислити суму додатних елементів масиву *A*.
34. Елементи масиву $A=(a_i)$, $(i=1,2,\ldots,n)$ задаються так:

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

35. Обчислити

$$z = \begin{cases} -1, & \text{якщо сума квадратів від'ємних елементів масиву менша за суму додатних,} \\ 1, & \text{в іншому випадку.} \end{cases}$$

36. Елементи масиву $A=(a_i)$, задаються так: $a_i = \frac{(i-1)^2}{2i^2-1} + i!\cdot\sin ix$ $(i=1,2,\ldots,n)$.

Побудувати масив $B=(b_i)$ так: $b_i = \begin{cases} a_1\cdot a_2\cdot\ldots\cdot a_i, & \text{якщо } a_i<0, \\ |a_1|+2|a_2|+\ldots+i|a_i|, & \text{в іншому випадку.} \end{cases}$

37. Елементи масиву $A=(a_i)$, задаються так:

$$a_i = 1!\cdot\sin b + 2!\cdot\sin 2b + \ldots i!\cdot\sin ib, \ (i=1,2,\ldots,n).$$

38. Знайти $\min\{a_1\cdot a_2, a_2\cdot a_3, a_3\cdot a_4, \ldots, a_{n-1}\cdot a_n\}$

39. Елементи масиву $A=(a_i)$ $(i=1,2,...,n)$ задаються так:

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

40. Знайти середнє арифметичне всіх елементів масиву, які потрапляють у про міжок $(b, c]$.

41. Побудувати масив $A=(a_i)$, елементи якого задаються формулою:

42. $a_i = \sin x \cos x - \sin 2x \cdot \cos(1 \cdot 2)x + \cdots + (-1)^{i+1} \sin ix \cdot \cos(1 \cdot 2 \cdot \ldots \cdot i)x, (i = 1,2,...,n),$

43. де *x* вводиться з клавіатури. Знайти порядковий номер найменшого елемента ма сиву *A.*

44. Побудувати масив $A=(a_i)$, елементи якого задаються формулою:

1. $a_i = \dfrac{-1 \cdot 2 \cdot (-3) \cdot \ldots \cdot (-1)^i i}{i}, \quad (i = 1,2,...,n).$

45. Обчислити суму додатних елементів масиву *A* і вивести її на форму.

46. Побудувати масив $A=(a_i)$, елементи якого задаються формулою:

47. $a_i = (\sin x + \cos x) + 2(\sin 2x + \cos 2x) + \cdots + i(\sin ix + \cos ix), \quad (i = 1,2,...,n).$

48. Знайти найбільший елемент масиву *A.*

49. Побудувати масив $A=(a_i)$, елементи якого задаються формулою:

$$a_i = \frac{1 + 1/2 + \cdots + 1/i}{i!}, \quad (i = 1,2,...,n).$$

50. Вивести добуток елементів масиву *A* з парними індексами.

51. Побудувати масив $X=(x_k)$, елементи якого задаються формулою:

$$x_1 = a,\ x_k = \frac{2 - x_{k-1}^2}{b}, \quad (k = 2,3...,n),$$

52. де *a, b* – вводяться з клавіатури. Знайти середнє арифметичне значення від'єм них елементів масиву *X.*

53. Побудувати масив $V=(v_i)$, елементи якого задаються формуло ю:

$$v_1 = a,\ v_2 = b,\ v_i = \frac{1}{2}\left(v_{i-1} + \frac{x}{v_{i-2}}\right), \quad (i = 3,4,...,n),$$

54. де *a, b, x* – вводяться з клавіатури. Обчислити добуток елементів масиву *V* з парними індексами

55. Побудувати масив $A=(a_i)$, елементи якого задаються формулою:

$$a_1 = a_2 = b, \quad a_i = a_{i-2} + \frac{a_{i-1}}{2^{i-1}}, \quad (i = 3,4,...,n),$$

56. де *b* – вводиться з клавіатури. Вивести суму элемент ів масиву *A* з непарними індексами.

57. Побудувати масив $A=(a_i)$, елементи якого задаються формулою:

$$a_1 = a_2 = x \quad a_i = 2a_{k-1}^2 + a_{k-2}, \quad (i = 3,4,...,n),$$

58. де *x* – вводиться з клавіатури. Знайти середнє геометричне чисел $|a_1|, |a_1 a_2|, \ldots$ $|a_1 a_2 \ldots a_n|$, де $|a|$ – модуль числа *n*.

59. Побудувати вектор $\bar{b}$, який містить тільки додатні координати заданого вектора $\bar{a}\,(a_1, a_2, ..., a_n)$, вивести його розмірність та довжину

60. По заданому вектору $\bar{X}\,(x_1, x_2, ..., x_n)$ побудувати вектор $\bar{Y}\,(y_1, y_2, ..., y_n)$, де

$$y_i = \begin{cases} x_i, & якщо\ x_i \geq 0, \\ x_i^{-1}, & якщо\ x_i < 0. \end{cases}$$

61. Задані два вектори $\bar{a}\,(a_1, a_2, ..., a_n)$ і $\bar{b}\,(b_1, b_2, ..., b_n)$. Побудувати новий вектор $\bar{c}\,(c_1, c_2, ..., c_n)$, у якого $c_i = \min(a_i, b_i)\ (i = 1,2,...,n)$ і знайти його довжину.

62. Дано два вектори $x, y \in R^n$. З'ясувати, чи паралельні вони.

63. Дано два вектори $x, y \in R^n$. З'ясувати, чи є вони перпендикулярними.

64. Дано дві числові послідовності, кожна з яких містить n елементів. Отримати S_1 – суму чисел, які стоять на непарних позиціях в першій послідовності, і S_2 – суму чисел на парних позиціях в другій.

65. Задані цілі числа $a, x_1, x_2, ..., x_n$. Якщо в послідовності $x_1, x_2, ..., x_n$ є хоча б один член, який дорівнює a, то обчислити суму всіх членів, які йдуть за першим таким числом, у протилежному випадку вивести 10.

66. Заданий вектор $\bar{a}\,(a_1, a_2, ..., a_n)$. Побудувати та вивести вектор $\bar{b}\,(b_1, b_2, ..., b_n)$, i-а координата якого дорівнює середньому-арифметичному перших i координат вектора $\bar{a}$:

$$b_i = \frac{a_1 + a_2 + ... + a_i}{i}.$$

67. Дано послідовність слів, які зберігаються в масиві рядкового типу. Визначити в якому з слів найбільше букв «а».

2. Багатовимірні масиви

Багатовимірними масивами є масиви, в яких для ідентифікації елемента необхідно вказати більше одного індексу. Найбільш використовуваними є двовимірні масиви, тому їх буде детально розглянуто. З багатовимірними масивами маніпуляції виконуються аналогічно.

2.1. Статичні двовимірні масиви фіксованого розміру та фіксованого типу

Раніше було зазначено, що елементи масиву можуть буди довільного типу, в тому числі і масивами. У випадку, якщо маємо одновимірний масив, елементами якого є також одновимірні масиви (однакової фіксованої довжини) то кажуть, що маємо двовимірний масив, для ідентифікації елементів якого необхідно вказати два індекси. Оскільки двовимірні масиви часто асоціюються з двовимірними таблицями і матрицями, то прийнято вважати, що перший індекс позначає номер рядка, а другий – номер стовпця.

Є дві форми опису статичного двовимірного масиву:

<ім'я масиву> : array[<діапазон зм. індексу1>] of
array[<діапазон зм. індексу2>] of <тип елементів>;

або

<ім'я масиву> : array [<діапазон зм. індексу1> , <діапазон зм. індексу2>] of
<тип елементів>;

Обидві форми описання описують один і той же самий двовимірний масив, тобто є ідентичними. Але, як правило, використовується друга форма опису, оскільки вона є більш лаконічною.

Приклад. Описані типи

```
TYPE
    TMatrix = array[1..30] of array [1..20] of Real;
```

та

```
TYPE
    TMatrix = array[1..30,1..20] of Real;
```

є ідентичними. Обидва типи описують двовимірний масив, який складається з 30 рядків та 20 стопвців, або ж одновимірний масив, що складається з 30

елементів, які є одновимірними масивами з 20 елементів типу Real. Так змінну A

```
VAR
    A : TMatrix;
```

можна відобразити у вигляді матриці

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1,20} \\ a_{21} & a_{22} & \dots & a_{2,20} \\ \dots & \dots & \ddots & \dots \\ a_{30,1} & a_{30,2} & \dots & a_{30,20} \end{pmatrix}.$$

Наведемо ще декілька прикладів опису типу двовимірних масивів.

Приклад.

```
CONST n=10;
      m=15;
TYPE
    TMatrix  = array[1..n,2..m] of string;
    TMatrix1 = array[0..30,2..20,1..24] of real;
    TMatrix2 = array['a'..'f','c'..'r'] of real;
```

Для двовимірних масивів актуальними є обмеження, введені для одновимірних статичних масивів:

1) елементи масиву можуть бути довільного, але тільки одного типу;
2) межі зміни індексів можуть бути тільки сталими величинами;
3) діапазони зміни індексів не можуть змінюватися під час виконання програми;
4) значення індексів не повинно виходити за межі визначених діапазонів;
5) масиви потрібно вводити/виводити поелементно, тобто кожен елемент окремо.

Отже, при використанні статичних двовимірних масивів, як і одновимірних, межі зміни діапазону індексів повинні бути константами, і діапазон не може змінюватися під час виконання програми. Тому в залежності від задачі при описі масиву необхідно вказати гранично можливі діапазони зміни індексів. Для збереження ж справжніх меж діапазонів зміни індексів описують додаткові змінні, значення яких не можуть виходити за вказані гранично можливі.

Приклад. Необхідно зберігати покази температури озера. Вимірювання проводяться рівномірно протягом певної кількості днів.

Позначимо n – кількість днів, протягом яких здійснювались виміри, а m – кількість щоденних вимірів температури озера. Для збереження вимірів можна використати двовимірний масив. Опишемо тип цього статичного двовимірного масиву, припускаючи, що $n \le 60$, а $m \le 15$.

```
TYPE
    TTemperature = array[1..60,1..15] of Real;
VAR
    T: TTemperature;
    n,m:integer;
```

Змінну T можна відобразити у вигляді таблиці

№ виміру / № дня	1	2	…	15
1	$T_{1,1}$	$T_{1,2}$	…	$T_{1,15}$
2	$T_{2,1}$	$T_{2,2}$	…	$T_{2,15}$
…	…	…	…	…
60	$T_{60,1}$	$T_{60,2}$	…	$T_{60,15}$

Тут $T_{i,j}$ – температура, що одержана i-того дня при j-товому вимірюванні.

Для ідентифікації елемента двовимірного масиву необхідно вказати ім’я масиву та два індекси елемента: перший індекс – номер рядка, другий – номер стовпця. Є дві еквівалентні форми ідентифікації елемента масиву:

```
<ім’я масиву>  [ <індекс1> ]  [ <індекс2> ]
```

та

```
<ім’я масиву>  [ <індекс1> , <індекс2> ]
```

Як правило, використовують другу форму, оскільки вона є більш лаконічною.

Наприклад. Обидва записи

```
A[2,7]
```

та

```
A[2][7]
```

є допустимими і еквівалентними. Вони ідентифікують елемент двовимірного масиву A, який знаходиться на перетині 1-го рядка і 7-го стовпця.

Представлення статичного двовимірного масиву в пам'яті ЕОМ

Для збереження статичних двовимірних масивів, як і для одновимірних, в пам'яті ЕОМ використовується послідовне представлення. При цьому наперед виділяється ділянка оперативної пам'яті величиною

<кількість рядків> ***<кількість стовпців>** * < **розмір одного елемента масиву** >

і елементи розміщуються послідовно, один за одним. Спочатку елементи першого рядка, потім другого, третього і т.д.

Приклад.

```
TYPE   TMatrix = array[1..3,1..2] of real;
VAR    A: TMatrix;
```

Змінну A

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{pmatrix}.$$

у пам'яті ЕОМ буде розміщено наступним чином:

								Комірки
…	A[1,1]	A[1,2]	A[2,1]	A[2,2]	A[3,1]	A[3,2]	…	**Імена комірок**
	Рядок 1		Рядок 2		Рядок 3			
	Область масиву A							

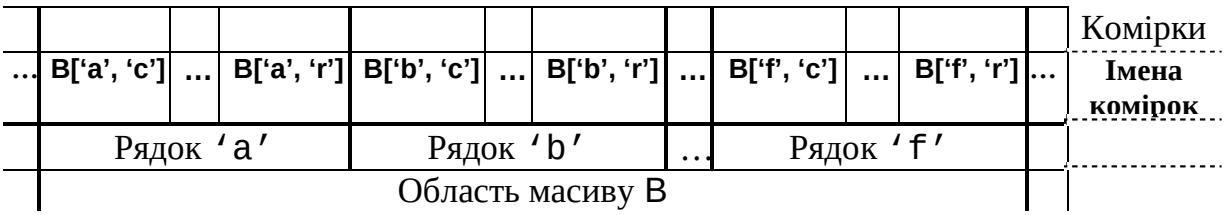
Приклад.

```
TYPE  TMatrix2 = array['a'..'f','c'..'r'] of real;
VAR   B: TMatrix2;
```

Змінну B

$$B = \begin{pmatrix} b_{a,c} & b_{a,d} & \dots & b_{a,r} \\ b_{b,c} & b_{b,d} & \dots & b_{b,r} \\ \dots & \dots & \ddots & \dots \\ b_{f,c} & b_{f,d} & \dots & b_{f,r} \end{pmatrix},$$

у пам'яті ЕОМ буде розміщено наступним чином



Введення елементів масиву

Двовимірний масив відноситься до структурованих типів даних, тому введення і виведення масивів повинно здійснюватися поелементно. Як правило, введення елементів масиву здійснюється за допомогою двох циклів. У першому змінюється перший індекс (номер рядка), а у вкладеному другому – другий індекс (номер стовпця). Загальний вигляд циклу введення елементів масиву може виглядати наступним чином:

```
for <параметр1> := <поч.зн.інд.1> to <кін. зн.інд.1> do
begin
  for <параметр2> := <поч.зн.інд.2> to <кін. зн.інд.2> do
   begin
       write('<ім"я мас.>[ ', <параметр1>, ',' , <параметр2>,' ]= ');
       readln(<ім"я мас.> [ <параметр> , <параметр2> ] );
   end;
end;
```

Приклад.

```
for  i := 1  to  n  do
begin
       for  j := 1  to  m  do
       begin
              write(' A [ ', i , ',' , j ,  ' ]= ');
              readln( A [ i , j ] );
       end;
end;
```

Приклад.

```
for  i := 'a'  to  'f'  do
begin
       for  j := 'c'  to  'r'  do
       begin
              write(' B [ ', i , ',' , j , ' ] = ');
              readln( B [  i , j ] );
       end;
end;
```

Виведення елементів масиву

Як і введення, виведення елементів масиву також здійснюється поелементно. Загальний вигляд циклу виведення елементів двовимірного масиву може виглядати наступним чином:

```
for <параметр1> := <поч.зн.інд.1> to <кін. зн.інд.1> do
begin
  for <параметр2> := <поч.зн.інд.2> to <кін. зн.інд.2> do
   begin
      writeln('<ім"я мас.>[ ', <параметр1>, ',' , <параметр2>,' ]= ',
            <ім"я мас.> [ <параметр> , <параметр2> ] );
   end;
end;
```

Приклад.

```
for  i := 1  to  n  do
begin
      for  j := 1  to  m  do
      begin
            writeln(' A [ ', i , ',' , j ,  ' ]= ', A [ i , j ] );
      end;
end;
```

Приклад.

```
for  i := 'a'  to  'f'  do
begin
      for  j := 'c'  to  'r'  do
      begin
            writeln(' B [ ', i , ',' , j , ' ] = ',  B [  i , j ] );
      end;
end;
```

Наведемо приклад задачі, розв'язаної з використанням двовимірного масиву.

Приклад. Дано матрицю $A_{n\times m}$. Необхідно помножити кожен елемент матриці на її найбільший елемент.

Розв'язання

Будемо вважати, що $n \le 25$, а $m \le 30$.

```
PROGRAM Matr;
{$APPTYPE CONSOLE}
    TYPE TMatrix= array[1..25,1..30] of real;
    VAR A: TMatrix;
        n,m, i,j :integer;
        max:real;
BEGIN
{------------- Введення розмірності масиву --------}
```

```
write('Введіть кількість рядків   n ='); readln(n);
write('Введіть кількість стовпців m ='); readln(m);
{------------  Введення масиву A --------}
    writeln('Введіть елементи масиву A');
    for i:=1 to n do
  begin
      for j:=1 to m do
        begin
          write('A[',i,',',j,']= '); readln(A[i,j]);
        end;
  end;
{===  Знаходження найбільшого елемента масиву A ======}
    max:=A[1,1];
    for i:=1 to n do
  begin
      for j:=1 to m do
        begin
        if A[i,j]>max then
         max:=A[i,j];
        end;
  end;
 {====  Множення матриці A на max ====}
    for i:=1 to n do
  begin
      for j:=1 to m do
        begin
         A[i,j]:=max*A[i,j];
        end;
   end;
    {-------- Виведення елементів масиву A ----------}
  writeln('--- Результат ---');
    for i:=1 to n do
  begin
      for j:=1 to m do
      begin
          writeln('  A[',i,',',j,']= ',A[i,j]:6:2);
      end;
  end;
    readln;
END.
```

Результат роботи програми

```
Введіть кількість рядків   n = 2
Введіть кількість стовпців m = 3
Введіть елементи масиву A
A[1,1]= 2
```

```
A[1,2]= 1
A[1,3]= 3
A[2,1]= 5
A[2,2]= 2
A[2,3]= 1
--- Результат ---
  A[1,1]=  10.00
  A[1,2]=   5.00
  A[1,3]=  15.00
  A[2,1]=  25.00
  A[2,2]=  10.00
  A[2,3]=   5.00
```

Копіювання статичних двовимірних масивів

Для копіювання значень з одного двовимірного масиву в інший двовимірний масив того ж типу, як і у випадку з одновимірними масивами, можна використати оператор присвоювання

<Масив 1> := < Масив 2> ;

Приклад. Нехай у програмі описано два одновимірних статичних масиви

```
Type TMatrix = array[1..3,1..2] of real;
VAR A,B: TMatrix;
```

Тоді після введення елементів масиву A, їх значення можна скопіювати у масив B, використавши оператор присвоювання

```
B:=A;
```

Двовимірні масиви-константи

При створенні програм є можливість описання типізованих двовимірних констант-масивів. При цьому необхідно поступово у дужках вказати список значень першого рядка, потім другого і т.д.

```
CONST
<ім'я масиву> : array [<діапазон зм. індексу1> , <діапазон зм. індексу2> ] of
 <тип елементів>=((<список знач.1>),(<список знач.2>),…,(<списо знач>N));
```

Приклад.

```
CONST
A:array[1..2,1..3] of real =((1.5,1.2,0.6),(0,2.1,3.5));
D:array['0'..'1','0'..'4'] of char=
    (('0','1','2','3'),('2','3','4','5'));
D1:array['0'..'1','0'..'4'] of char=(('0123'),('2345'));
```

2.2. Динамічні двовимірні масиви з фіксованим типом елементів

Як і для одновимірних статичних масивів, суттєвим недоліком двовимірних статичних масивів є заборона на зміну діапазонів зміни індексів. Тому на практиці частіше використовують динамічні двовимірні масиви фіксованого типу, для яких діапазони зміни індексів визначаються під час виконання програми.

При описі динамічних двовимірних масивів фіксованого розміру кількість елементів масиву не зазначається, а вказується тільки тип його елементів. Наведемо загальний вигляд їх опису:

```
TYPE
    <ім'я типу масиву> = array of array of <тип елементів>;
```

Приклад.

```
TYPE   TMatrixA = array of array of byte;
       TMatrixB = array of array of real;
       TMatrixC = array of array of string;
VAR    A: TMatrixA;
       B: TMatrixB;
       C: TMatrixC;
```

Змінні типу динамічний масив є вказівниками. Пам'ять для елементів масиву виділяється під час виконання програми з використанням процедури SetLength, загальна форма виклику якої наступна:

SetLength(<ім'я масиву>,<кількість рядків>,<кількість стовпців>);

При цьому буде виділено пам'ять для збереження двовимірного масиву, у якому вказана кількість рядків і кожен з рядків містить вказану кількість

стовпців. Адреса початку пам'яті зберігається у вказівнику – імені динамічного масиву. Нумерація рядків і стовпців масиву починається від 0.

Приклад.

```
SetLength( A , 10,15);      //  A10×15 .
SetLength( B , 25,30);      //  B25×30 .
SetLength( C , 3,7);        //  C3×7
```

Індекс першого рядка масиву можна визначати за допомогою функції Low, загальна форма виклику якої наступна:

Low (<ім'я динамічного масиву>);

Індекс останнього рядка масиву можна визначати за допомогою функції High, загальна форма виклику якої наступна:

High (<ім'я динамічного масиву>);

Номер першого стовпця можна визначити, застосувавши функц ію Low до якогось із рядків масиву:

Low (<ім'я динамічного масиву>[< номер рядка масиву >]);

Номер останнього стовпця можна визначити, застосувавши функцію High до якогось із рядків масиву:

High (<ім'я динамічного масиву>[< номер рядка масиву >]);

Приклад.

```
    TYPE   TMatrixA = array of array of Byte;
    VAR    A: TMatrixA;
           FirstRow,LastRow, FirstCol,LastCol:Integer;
BEGIN
    SetLength( A , 10,15);
    FirstRow:= Low( A );          // FirstRow = 0
    LastRow:= High( A );          // LastRow = 9
    FirstCol:= Low( A[0] );       // FirstCol = 0
    LastCol:= High( A[0]);        // LastCol = 14

End.
```

Звертання до елементів масиву можна здійснювати як і при використанні статичних масивів:

<ім'я масиву> [<індекс1> , <індекс2>]

Приклад.

```
A [ 0,1 ] := 2;
B [ 4 ,3] := 2.4;
C [ 3, 2 ] := ‘ Мир ’ ;
```

Проілюструємо використання динамічних одновимірних масивів на прикладі.

Приклад. Дано матриці $A_{n\times m}$ та $B_{n\times m}$. Знайти матрицю $C = A + B$.

Розв’язання

Для збереження матриць використаємо однойменні двовимірні динамічні масиви фіксованого типу A, B і C.

```
PROGRAM P_2_1D;
{$APPTYPE CONSOLE}
    Type TVector = array of array of real;
    VAR  A,B,C: TVector;
         n,m,i,j:integer;
BEGIN
write('Введіть кількість рядків n ='); readln(n);
write('Введіть кількість стовпців m ='); readln(m);
{------------  Введення масиву A --------}
   SetLength(A,n+1,m+1);
    writeln('Введіть елементи масиву A');
    for i:=1 to n do
      begin
    for j:=1 to m do
        begin
        write('A[',i,',',j,']= '); readln(A[i,j]);
      end;
      end;
{------------  Введення масиву B --------}
    writeln('Введіть елементи масиву B');
   SetLength(B,n+1,m+1);
    for i:=1 to n do
      begin
    for j:=1 to m do
        begin
        write('B[',i,',',j,']= '); readln(B[i,j]);
      end;
      end;

{============ Знаходження вектора C ============}
   SetLength(C,n+1,m+1);
    for i:=1 to n do
      begin
```

```
    for j:=1 to m do
        begin
        C[i,j]:= A[i,j]+B[i,j];
      end;
      end;
{--------------- Виведення масиву C ----------}
    writeln('Масив C');
    for i:=1 to n do
      begin
    for j:=1 to m do
        begin
        writeln('C[',i,',',j,']= ',C[i,j]:6:2);
      end;
      end;
    readln;
END.
```

Результати роботи програми

```
Введіть кількість рядків n = 2
Введіть кількість стовпців m = 3
Введіть елементи масиву A
A[1,1]= 2
A[1,2]= 1
A[1,3]= 3
A[2,1]= 4
A[2,2]= 1
A[2,3]= 5
Введіть елементи масиву B
B[1,1]= 6
B[1,2]= 2
B[1,3]= 1
B[2,1]= 4
B[2,2]= 3
B[2,3]= 0
Масив C
C[1,1]=   8.00
C[1,2]=   3.00
C[1,3]=   4.00
C[2,1]=   8.00
C[2,2]=   4.00
C[2,3]=   5.00
```

Динамічні масиви з різною довжиною рядків

Ще однією перевагою динамічних двовимірних масивів є те, що кожен рядок такого масиву може мати різну кількість елементів. Зрозуміло, що в цьому випадку пам'ять для кожного рядка необ хідно виділяти окремо.

Приклад. Створимо двовимірний динамічний масив елементів дійсного типу з кількістю елементів у нульовому рядку – 5 елементів, першому – 3 елементи, другому – 4 елементи.

```
TYPE      TMatrix = array of array of Real;
VAR       A : TMatrix;
BEGIN
     SetLength(A,3);                    // Створення 3-х рядків
          SetLength( A[0] , 5 );        // 0-вий рядок – 5 елементів
          SetLength( A[1] , 3 );        // 1-вий рядок – 3 елементів
          SetLength( A[2] , 4 );        // 2-вий рядок – 4 елементів
END.
```

При реалізації циклів кількість елементів і-го рядка можна визначати за допомогою функцій Low та High. Такий цикл може виглядати наступним чином:

```
    for i:=Low(A) to High(A) do
      begin
        for j:=Low(A[i]) to High(A[i]) do
         begin
          . . . . . . . . . . . . . .
         end;
      end;
```

Проілюструємо дану можливість на прикладі.

Приклад. У *n* групах студентів є члени футбольної команди (кількість футболістів у групах може бути різною). Знайти вік найстаршого футболіста.

Розв'язання

Для збереження віку футболістів використа ємо динамічний двовимірний масив A (тут A[i , j] – вік j-того футболіста з і-тої групи).

```
PROGRAM P_2_1DD;
{$APPTYPE CONSOLE}
    Type TVector = array of array of integer;
    VAR  A: TVector;
         n,max,i,j,k:integer;
```

```
BEGIN
write('Введіть кількість груп : '); readln(n);
{------------  Введення масиву A --------}
   SetLength(A,n+1);     //Створення рядків
for i:=1 to n do
  begin
      write('Введіть кількість футболістів у групі ',i,' : ');
        readln(k);
      SetLength(A[i],k+1); //Створення елементів і-го рядка
      writeln('Задайте вік футболістів у групі  ',i);
    for j:=1 to k do
     begin
        write('Вік футболіста № ',j,' : ');
         readln(A[i,j]);
     end;
  end;
{-----  Знаходження віку найстаршого футболіста ------}
   max:=A[1,1];
    for i:= Low(A)+1 to High(A) do
      begin
        for j:=Low(A[i])+1 to High(A[i]) do
        begin
          if A[i,j]>max then
          max:=A[i,j];
        end;
      end;
    writeln('Найстаршому футболісту  ',max,' років');
 readln;
END.
```

Результат роботи програми

```
Введіть кількість груп : 3
Введіть кількість футболістів у групі  1 : 4
Задайте вік футболістів у групі   1
Вік футболіста №  1 : 23
Вік футболіста №  2 : 25
Вік футболіста №  3 : 26
Вік футболіста №  4 : 25
Введіть кількість футболістів у групі  2 : 2
Задайте вік футболістів у групі   2
Вік футболіста №  1 : 25
Вік футболіста №  2 : 24
Введіть кількість футболістів у групі  2 : 3
Задайте вік футболістів у групі   3
Вік футболіста №  1 : 24
```

```
Вік футболіста №  2 : 22
Вік футболіста №  3 : 25
Найстаршому футболісту 26 років
```

Копіювання динамічних двовимірних масивів

Копіювання динамічних двовимірних масивів, які і одновимірних здійснюється з використанням функції Copy:

<Динам. масив 1>:=Copy(<Динам. масив 2>);

Приклад.

```
B:=Copy(A);
```

Дана функція виділяє для масиву B стільки ж пам'яті, скільки займає масив A, а потім копіює значення з масиву A у масив B.

2.3. Динамічні двовимірні масиви змінного типу елементів. Variant-масиви

Як і для одновимірних динамічних масивів, о сновою динамічних двовимірних масивів є тип Variant. Аналогічною є також схема їх використання:

1) підключаємо модуль Variants;

Uses Variants;

2) описуємо змінну-масив, як змінну типу Variant;

Var <ім'я масиву> : Variant;

3) виділяємо пам'ять для масиву (при цьому вказуємо початкове, кінцеве значення індексу та тип елементів масиву).

<ім'ямасиву>:=**VarArrayCreate([**<поч.зн.інд.1>,<кін.зн.інд1>,<поч.зн.інд.2>,<кін.зн.інд2>**]**,
<тип елем.>);

<поч.зн.інд.1>,<кін.зн.інд1> визначають діапазон зміни номерів рядків, а <поч.зн.інд.2>,<кін.зн.інд2> визначають діапазон зміни ном ерів стовпців.

Приклад.

```
    Uses Variants;
    VAR A,B: Variant;
        n:integer;
    . . . . . .
BEGIN
    . . . . . .
```

```
  A:=VarArrayCreate([0,n,0,n],varInteger);
                                        {(n+1) рядок, (n+1) стовпець}
  B:=VarArrayCreate([1,7,1,9],varDouble);
                                        {7 рядків, 9 стовпців}
  V:=VarArrayCreate([1,10,1,15], varVariant);
                                        {10 рядків, 15 стовпців}
     . . . . . .
END.
```

Проілюструємо використання двовимірних Variant-масивів на раніше розглядуваному прикладі.

Приклад. Дано матриці $A_{n\times m}$ та $B_{n\times m}$. Знайти матрицю $C = A + B$.

Розв'язання

```
PROGRAM SumMatrix;
{$APPTYPE CONSOLE}
    USES Variants;
    VAR A,B,C: Variant;
        n,m,i,j:integer;
        s:string;
BEGIN
 write('Введіть кількість рядків n ='); readln(n);
 write('Введіть кількість стовпців m ='); readln(m);
{------------  Введення масиву A --------}
   A:=VarArrayCreate([1,n,1,m],varInteger);
    writeln('Введіть елементи масиву A');
    for i:=1 to n do
      begin
    for j:=1 to m do
        begin
        write('A[',i,',',j,']= '); readln(s);
       A[i,j]:=s;
      end;
      end;
{------------  Введення масиву B --------}
    writeln('Введіть елементи масиву B');
   B:=VarArrayCreate([1,n,1,m],varInteger);
    for i:=1 to n do
      begin
    for j:=1 to m do
        begin
        write('B[',i,',',j,']= '); readln(s);
       B[i,j]:=s;
      end;
      end;
```

```
{============= Знаходження вектора C =============}
   C:=VarArrayCreate([1,n,1,m],varInteger);
    for i:=1 to n do
      begin
    for j:=1 to m do
        begin
        C[i,j]:= A[i,j]+B[i,j];
      end;
      end;
{--------------- Виведення масиву C ----------}
    writeln('Масив C');
    for i:=1 to n do
      begin
    for j:=1 to m do
        begin
        writeln('C[',i,',',j,']= ',C[i,j]);
      end;
      end;
    readln;
END.
```

Результат роботи аналогічний до попереднього прикладу.

Динамічні двовимірні масиви з рядками різної довжини і різного типу

Як і при використанні динамічних масивів фіксованого типу, при використанні динамічних двовимірних Variant-масивів є можливість створення двовимірних масивів з рядками різної довжини та різного типу.

Приклад. Створити двовимірний масис з 3-х рядків. У першому рядку 4 елементи цілого типу, у другому 7 елементів дійсного типу, у третьому 2 елементи рядкового типу.

```
    Uses Variants;
    VAR A: Variant;
    . . . . . .
BEGIN
    . . . . . .
     A:=VarArrayCreate([1,3],varVariant);

        A[1] := VarArrayCreate([1,4], varInteger);

        A[2] := VarArrayCreate([1,7], varDouble);

        A[3] := VarArrayCreate([1,2], varOleStr);
          . . . . . .
END.
```

Питання для самоконтролю

1. *Що таке двовимірний масив?*
2. *Як задаються діапазони зміни індексів?*
3. *Як зберігаються двовимірні масиви в пам'яті ЕОМ?*
4. *Які особливості введення двовимірного масиву?*
5. *Які особливості виведення двовимірного масиву?*
6. *Як описуються двовимірні масиви-константи?*
7. *Як описуються динамічні двовимірні масиви фіксованого типу?*
8. *Чим відрізняються статичні двовимірні масиви від динамічних двовимірних масивів фіксованого типу?*
9. *Яка особливість змінної динамічного двовимірного масиву фіксованого типу?*
10. *Які межі зміни індексів у динамічних двовимірних масивах фіксованого типу?*
11. *Яка процедура використовуються для виділення пам'яті для динамічних двовимірних масивів фіксованого типу?*
12. *Як визначити максимальне та мінімальне значення індексу рядка динамічного двовимірного масиву фіксованого типу?*
13. *Як визначити максимальне та мінімальне значення індексу стовпця динамічного двовимірного масиву фіксованого типу?*
14. *Як можна описати динамічний двовимірний масив з різною кількістю елементів у рядку?*
15. *Як здійснюється копіювання значень динамічного двовимірного масиву фіксованого типу?*
16. *Що таке динамічний двовимірний масив змінного типу?*
17. *Як виділяється пам'ять для динамічних масивів змінного типу?*
18. *Якого типу є змінна-динамічний двовимірний масив змінного типу?*
19. *Як описати Variant-масив з різною кількістю елементів у рядку та різним типом елементів?*

Завдання для самостійної роботи

1. Створити програму, для знаходження детермінанта квадратної матриці $A(2\times 2)$.

2. Дано матрицю $A(m\times n)$ та вектор $x(n)$. Знайти вектор $y = Ax$.

3. Дано матриці A, B, C розмірності $(n\times n)$ знайти $A + B + C$.

4. Дано матрицю $A(n\times n)$. Знайти суму елементів, розташованих нижче головної діагоналі.

5. У квадратній матриці $A(n\times n)$ елементи головної діагоналі замінити найбільшим елементом цієї матриці.

6. У квадратній матриці знайти суму від'ємних елементів, які розташовані нижче головної діагоналі.

7. Побудувати матрицю A, елементи якої задаються формулою:

$$a_{ij} = \begin{cases} 1\cdot 2\cdot 3\cdot \ldots \cdot j, & \text{якщо} \quad i\cdot j - \text{парне} \\ 1+2+\ldots+i, & \text{якщо} \quad i\cdot j - \text{непарне} \end{cases}, \quad i = \overline{1,n}, \quad j = \overline{1,m}.$$

8. Побудувати одновимірний масив (переписати матрицю в одновимірний масив).

9. Дано матрицю $A_{m\times n}$, та матриця B такої самої розмірності. Замінити всі нульові елементи матриці A відповідними елементами матриці B.

10. Побудувати квадратну матрицю A, елементи якої задаються формулою:

$$a_{ij} = \begin{cases} i + j, & \text{якщо } i\cdot j < 3 \\ -1+2+\ldots+(-1)^j j, & \text{якщо } i\cdot j \geq 3 \end{cases}, \quad i, j = \overline{1,n}.$$

11. Знайти максимальний елемент головної діагоналі і добуток елементів того рядка, у якому він знаходиться.

12. Дано дійсну матрицю розмірності $n\times n$, всі елементи якої різні. Знайти скалярний добуток i-го рядка і j-го стовпчика (i, j задаються користувачем).

13. Побудувати квадратну матрицю A, елементи якої задаються формулою:

$$a_{ij} = \begin{cases} 1+2+\ldots+i, & \text{якщо } i + j - \text{парне} \\ 1^2+2^2+\ldots+j^2, & \text{якщо } i + j - \text{непарне} \end{cases} \quad i, j = \overline{1,n}.$$

14. Сформувати одновимірний масив *b*, *i*-ий елемент якого є добутком усіх елементів *i*-го стовпчика матриці *A*. Знайти найбільший парний елемент масиву *b*.

15. Побудувати матрицю *A*, всі елементи якої різні. Знайти середнє-арифметичне значення додатних елементів того рядка, в якому зна ходиться максимальний елемент матриці.

16. Побудувати прямокутну матрицю *A*, елементи якої задаються формулою:

$$a_{ij} = i(\sin(ix) + \cos(jx)), \quad i = \overline{1, N}, \quad j = \overline{1, M}.$$

17. Обчислити добуток елементів, добуток ін дексів яких менший за задане число *x***.**

18. Побудувати прямокутну матрицю *A*, елементи якої задаються формулою:

$$a_{ij} = j\cos(i^2 + n), \quad i, j = \overline{1, N}.$$

19. Обчислити суму елементів матриці *A*, сума індексів яких непарна.

20. Побудувати квадратну матрицю *A*, елементи якої задаються формулою:

$$a_{ij} = \sin\left(\left(i^2 - j^2\right)/n\right), \quad i, j = \overline{1, N}.$$

21. Знайти найбільший за модулем елемент матриці *A* та його індекси.

ЗМІСТ

Відповідальний за випуск: завідувач кафедрою системного аналізу і теорії оптимізації к. ф.-м. н., доц. Кузка О.І.

Автори: к. т. н., доц. Семйон І.В.,
к. ф.-м. н., доц. Чупов С.В.,
к. ф.-м. н., Брила А.Ю.,
к. пед. н., Апшай Н.І.

Рецензенти: к. ф.-м.н., доц. Мич І.А.,
к. ф.-м.н., доц. Гече Ф.Е.

ВИКОРИСТАННЯ МАСИВІВ У DELPHI

Методичні вказівки до лабораторних робіт для студентів І І-го курсу математичного факультету спеціальності "прикладна мате матика"