

Основи алгоритмізації та програмування

## *Табличні величини*

Складання й реалізація програм  
опрацювання табличних величин

# **Знаходження суми й добутку елементів таблиць**



# Повторити теоретичний матеріал

---

Поняття масиву (таблиці);

Вказівки повторення з передумовою, з післяумовою, з параметром;

Способи опису табличних величин;

Способи формування табличних величин;

Способи виводу елементів табличних величин

# Задачі

---

- **Задача №1.** Числовий ряд в реверсному вигляді. Сума й добуток елементів масиву.
- **Задача №2.** Одномірний масив з парними індексами.
- **Задача №3.** Барон Мюнхаузен, вийшов на екологічно чисте полювання.
- **Задача №4.** Середня група дитячого садочка на прогулянці.
- **Задача №5.** Гідрометеоцентр веде спостереження і прогнозує.
- **Задача №6.** Цікава сума й добуток елементів масиву.
- **Задача №7.** Цікава сума й добуток, а також частка елементів масиву.
- **Задача №8.** Сума і добуток елементів в двомірних масивах.
- **Задача №9.** Сума елементів бічної діагоналі в двомірному масиві.
- **Задача №10.** Модифікована двомірна таблиця.

**Задача №1.** Дано одновимірний масив цілих чисел  $A[i]$ , де  $i = 1, 2, \dots, n$ . Вивести елементи масиву у зворотному порядку. Знайти суму всіх чисел.

```
Program Revers_Suma;
Uses crt;
Var N, i, S:integer; {N - кількість елементів
    масиву, i - змінна циклу}
    A:array [1..100] of longint; {A - заданий
    масив}
Begin
Clrscr;
Write ('Введіть кількість елементів масиву
    (<100): ');
Readln(N);
For i:=1 to N do
    Begin
A[i]:=random(300); {Заповнення масиву
    випадковими числами}
Write(A[i]:5); {Виведення масиву на екран
    для контролю правильності роботи
    програми}
    End;
```

```
WriteLn; {Переведення курсору на
    наступний
    рядок}
For i:=N downto 1 do
    Begin
Write (A[i]:5);
    End;
WriteLn;
S:=0;
For i:=1 to N do
    Begin
S:=S+A[i];
    End;
WriteLn('S=', S);
ReadLn; {Затримка зображення на
    екрані}
End.
{Внесіть зміни в програму для
    знаходження добутку
    елементів масиву}
```

**Задача №2.** Дано одновимірний масив цілих чисел  $A[i]$ , де  $i = 1, 2, \dots, n$ .  
Вивести елементи масиву з парними індексами.

```
Program Parnist;  
  Uses crt;  
  Var N,i : word; {N - кількість  
    елементів масиву, i - змінна циклу}  
      A : array [1..100] of longint; {A –  
    заданий масив}  
Begin  
  Clrscr;  
  Write ('Введіть кількість елементів  
    масиву (<100): ');  
  Readln (N);  
  For i:=1 to N do  
  Begin  
    A[i]:=random(300); {Заповнення масиву  
    випадковими числами}  
    Write(A[i]:5); {Виведення масиву на  
    екран для контролю правильності  
    роботи програми}  
  End;
```

```
  Writeln; {Переведення курсору на  
    наступний рядок}  
  i:=2;  
  while i<=N do  
  Begin  
    Write(A[i]:5);  
    i:=i+2; {Змінна циклу змінюється на 2,  
    щоб вибрати тільки парні елементи}  
  End;  
  ReadLn; {Затримка зображення на  
    екрані}  
End.
```

*{Знайдіть суму або добуток елементів з парними індексами}*

**Задача №3.** Барон Мюнхаузен, вийшовши на екологічно чисте полювання, зарядив свою рушницю кісточками вишень. Після того як він вдало влучив поміж роги оленям, в яких влучило відповідно  $k_1, k_2, \dots, k_n$  кісточок, у них на головах вирости чудові молоді вишеньки. Скільки саджанців зміг подарувати барон Мюнхаузен садівникам-дослідникам?

```
Program Baron;  
Uses crt;  
Var N:word; {N - кількість оленів, тобто  
елементів масиву, у яких влучив  
Мюнхаузен}  
K:array[1..100] of longint; {K -  
зарезервований масив для зберігання  
кількості кісточок, що влучили в  
оленів}  
i,Sum:longint; {i - змінна циклу, Sum -  
загальна кількість кісточок, що  
влучили в оленів}  
Begin  
Randomize; {Ця процедура запускається  
з метою зробити числа генератора  
випадкових чисел ще більш  
"випадковими"}
```

```
Clrscr;  
Sum:=0; {На початку роботи програми  
Мюнхаузен  
ще ні в кого не влучив}  
Write ('Введіть кількість оленів, в яких влучив  
Мюнхаузен (<=100): ');  
Readln(N);  
For i:=1 to N do  
Begin  
K[i]:=random(50)+20; {Заповнення масиву  
випадковими числами в діапазоні від 20 до  
70}  
Write(K[i]:5); {Виведення масиву на екран  
для контролю правильності роботи програми}  
Sum:=Sum+K[i];  
End;  
Writeln; {Переведення курсору на новий рядок}  
Writeln ('Кількість нових саджанців ',Sum);  
ReadLn; {Затримка зображення на екрані}  
End.
```

**Задача №4.** Середню групу дитячого садочка вивели на прогулянку. Скільки дівчаток і скільки хлопчиків видно з-за паркану, якщо зріст хлопчиків задається у сантиметрах від'ємними числами, а дівчаток - додатними у вигляді цілих значень  $a_1, a_2, \dots, a_n$ ? Крім того, у всіх дівчаток на голівках зав'язані бантики заввишки 10 см, а висота паркану  $H$  см.

```
Program Children;
Uses crt;
Var N,H:word; {N - кількість дітей в
              дитсадочку, H - висота паркану}
A:array[1..100] of longint; {A - зарезервований
масив для зберігання зростів дітей}
i,Count_girl,Count_boy:longint;
{i - змінна циклу, Count_girl - кількість
  дівчаток, Count_boy - кількість хлопців}
Begin
Randomize;
Clrscr;
Count_girl:=0;
Count_boy:=0;
Write ('Введіть висоту паркану: ');
Readln(H);
Write ('Введіть кількість дітей в дитсадочку: ');
Readln(N);
```

```
For i:=1 to N do
Begin
A[i]:=random(300)-150; {Заповнення масиву
                      випадковими числами в діапазоні від -150
                      до +150}
Write(A[i]:5); {Виведення масиву на екран
для контролю правильності роботи програми}
if (A[i]<0) and (abs(A[i])>H)
then Count_Boy:=Count_Boy+1;
if (A[i]>0) and (A[i]+10>H)
then Count_Girl:=Count_Girl+1;
End;
Write ('Кількість хлопчиків, що видно з-за
паркану ');
Writeln (Count_Boy);
Write ('Кількість дівчаток, що видно з-за
паркану ');
writelnCount_girl);
ReadLN; {Затримка зображення на екрані}
End.
```

**Задача №5.** Дано дійсні числа  $a_{1951}, a_{1952}, \dots, a_{2000}$  - кількість опадів (в мм), що випали у місті за останні 50 років минулого століття. Обчислити середню кількість опадів за цей період і щорічне відхилення від середнього значення.

```
Program Rain;
Uses crt;
Var N,i:longint; {N - кількість елементів
                масиву, i
                - змінна циклу}
A:array[1951..2000] of real; {A - масив для
                зберігання кількості
                опадів у відповідному році}
B:array[1951..2000] of real; {B - масив для
                зберігання відхилення від середнього
                значення}
Begin
Randomize;
Clrscr;
Sum:=0;
For i:=1951 to 2000 do
Begin
A[i]:=random(500)/7; {Заповнення масиву
                випадковими дійсними числами}
Write(A[i]:8:2); {Виведення масиву на екран для
                контролю роботи програми}
Sum:=Sum+A[i];
End;
```

```
Sum:=Sum/50; {Знаходження середньої
                кількості опадів за рік}
Writeln;
Writeln ('Щорічні відхилення від
                середньої кількості опадів за період
                1951 - 2000 р.р. ');
For i:=1951 to 2000 do
Begin
B[i]:=Sum - A[i]; {Знаходження
                щорічного відхилення}
Write(B[i]:8:2); {Виведення результатів
                на екран}
End;
ReadLN; {Затримка зображення на
                екрані}
End.
```



**Задача №6.** Дано дійсні числа  $a_1, a_2, \dots, a_{30}, b_1, b_2, \dots, b_{30}$ .  
Обчислити  $(a_1 + b_{30})(a_2 + b_{29}) \dots (a_{30} + b_1)$ .

```
Program Suma_Dobutok;  
Uses crt;  
Var A,B: array[1..30] of real;  
{A,B - масиви для зберігання вхідних даних}  
i:byte; {i - змінна циклу}  
Rez:real; {Rez - результат обчислень}  
Begin  
  Randomize;  
  Clrscr;  
  Writeln ('Масив A:');  
  For i:=1 to 30 do  
  Begin  
    A[i]:=random(200)/7-random*15;  
    Write(A[i]:8:2);  
  End;
```

```
  Writeln ('Масив B:');  
  For i:=1 to 30 do  
  Begin  
    B[i]:=random*200-random*100;  
    Write(B[i]:8:2);  
  End;  
  Writeln;  
  Rez:=1;  
  {Початкове значення дорівнює 1, тому що  
  результат являється накопиченням  
  добутку}  
  For i:=1 to 30 do  
  Begin  
    Rez:=Rez*(A[i]+B[31-i]);  
  End;  
  Writeln ('Результат обчислень = ',Rez:8:2);  
  ReadLn; {Затримка зображення на екрані}  
End.
```

**Задача №7.** Дано дійсні числа  $a_1, a_2, \dots, a_{29}, b_1, b_2, \dots, b_{30}$ . Обчислити

$$\frac{a_1 b_1 + a_3 b_3 + \dots + a_{29} b_{29}}{a_2 b_2 + a_4 b_4 + \dots + a_{30} b_{30}}$$

```
Program Suma_Dobut_Chastka;
Uses crt;
Var A,B:array[1..30] of real;
{A,B - масиви для зберігання вхідних даних}
i:byte; {i - змінна циклу}
R1,R2:real; {R1 - чисельник дроби, R2 -
           знаменник дроби}
Rez:real; {Rez - результат обчислень}
Begin
Randomize;
Clrscr;
Writeln ('Масив A:');
For i:=1 to 30 do
Begin
A[i]:=random(200)/7-random*15;
Write(A[i]:8:2);
End;
```

```
Writeln ('Масив B:');
For i:=1 to 30 do
Begin
B[i]:=random*200-random*100;
Write(B[i]:8:2);
End;
Writeln;
R1:=0;
R2:=0;
{Початкове значення дорівнює 0, тому що
   результат являється накопиченням суми}
For i:=1 to 15 do
Begin
R1:=R1 + (A[2*i-1]+B[2*i-1]);
R2:=R2 + (A[2*i]+B[2*i]);
End;
Rez:=R1/R2;
Writeln ('Результат обчислень = ',Rez:8:2);
ReadLn; {Затримка зображення на екрані}
End.
```

**Задача №8.** Дано натуральні числа  $n, m$  та матриці цілих чисел  $A_{ij}, B_{ij}$ , де  $i = 1, 2, \dots, n, j = 1, 2, \dots, m$ . Обчислити значення елементів матриці  $C_{ij}$ , якщо  $C_{ij} = A_{ij}(I+j) - B_{ij}(i^2+j^2)$ .

```
Program Suma_2Tabl;
Uses crt;
Const
n = 8;
m = 15;
Var A,B,C: array[1..n,1..m] of integer;
i,j:word; {i,j - змінні циклу}
Begin
Randomize;
Clrscr;
Writeln ('Масив A:');
For i:=1 to n do
Begin
For j:=1 to m do
begin
A[i,j]:=random(50)-random(20);
Write (A[i,j]:5);
end;
writeln;
End;
```

```
Writeln ('Масив B:');
For i:=1 to n do
Begin
For j:=1 to m do
begin
B[i,j]:=random(120)-random(80);
Write (B[i,j]:5);
end;
writeln;
End;
Writeln ('Результуючий масив C:');
For i:=1 to n do
Begin
For j:=1 to m do
begin
C[i,j]:=A[i,j]*(i+j)-B[i,j]*(i*i+j*j);
Write (C[i,j]:5);
end;
writeln;
End;
ReadLn; {Затримка зображення на екрані}
End.
```

**Задача №9.** Дано квадратну матрицю розмірності  $n$ .  
Надрукувати суму елементів бічної діагоналі.

---

```
Program Suma_Bichna;  
Uses crt;  
Const  
n = 10;  
Var A: array[1..n,1..n] of real;  
i,j:integer; {i,j - змінні циклу}  
Sum:real; {Sum - сума елементів бічної  
          діагоналі}  
Begin  
Randomize;  
Clrscr;  
{Заповнення масиву та виведення його на  
  екран}  
For i:=1 to n do  
Begin  
For j:=1 to n do  
begin  
A[i,j]:=random*50-random(80)/3;  
Write(A[i,j]:8:3);  
end;  
writeln;  
End;
```

```
Sum:=0; {Початкове значення суми}  
For i:=1 to n do  
Begin  
For j:=1 to n do  
begin  
if i + j = n+1  
then Sum:=Sum+A[i,j];  
end;  
End;  
Writeln ('Сума елементів бічної діагоналі  
  =',Sum:8:2);  
ReadLn; {Затримка зображення на  
  екрані}  
End.
```

**Задача №10.** Дано цілочислову прямокутну таблицю порядку  $n \times m$ . Усі елементи таблиці, менші за середнє арифметичне її значень, замінити на "-1", а більші - на "1".

```
Program Zamina;
Uses crt;
Const n = 9;
      m = 12;
Type
  Masiv = array[1..n,1..m] of integer;
Var A : Masiv;
    i,j : byte; {i,j - змінні циклу}
Sum,SA : real; {Sum - сума елементів таблиці, SA -
               середнє арифметичне}
Begin
  Randomize;
  Clrscr;
  Sum:=0; {Початкове значення суми}
  Writeln('Вихідний масив: ');
  For i:=1 to n do
  Begin
    For j:=1 to m do
    begin
      A[i,j]:=random(120)-random(65);
      Write(A[i,j]:5);
      Sum:=Sum+A[i,j]; {Накопичення суми елементів
                       масиву}
    end;
  end;
```

```
writeln;
End;
SA:=Sum/(n*m);
Writeln('Середнє арифметичне - ',SA:8:2);
Writeln('Результуючий масив: ');
For i:=1 to n do
Begin
  For j:=1 to m do
  begin
    if A[i,j] < SA then A[i,j]:=-1;
    if A[i,j] > SA then A[i,j]:=1;
    Write(A[i,j]:5);
  end;
  writeln;
End;
ReadLn; {Затримка зображення на екрані}
End.
```

## **Робота в середовищі програмування**

---

- Виконати програму і вивести результат програми на екран дисплея. Результат виконання записати в зошит.
- Зберегти програму на диску. Результат виконання записати в зошит.
- Відредагувати програму за завданнями вчителя.