

## **Лекція 10.**

### **Одновимірні масиви.**

### **Генерація псевдовипадкових чисел**

---

*Приклад.* З клавіатури вводиться  $N$  чисел.  
Знайти середнє арифметичне введених значень.

```
int main() {  
    int n, i;  
    double x, average = 0;  
    printf("n = "); scanf("%d", &n);  
    for (i = 0; i < n; i++) {  
        printf("a%d = ", i); scanf("%lf", &x);  
        average += x;  
    }  
    average /= n;  
    printf("Average = %f\n", average);  
    return 0;  
}
```

```
n = 10  
a0 = -5  
a1 = 6  
a2 = 3  
a3 = 10  
a4 = 3  
a5 = 1  
a6 = 5  
a7 = 1  
a8 = -3  
a9 = 5  
Average = 2.6
```

Часто виникає необхідність зберігати у пам'яті набори даних.

**Приклад.** З клавіатури вводиться  $N$  чисел. Знайти кількість чисел, які менші за середнє арифметичне введених значень.




Середнє арифметичне = 2.6

Кількість шуканих чисел: 4

---

*Оскільки у циклі на кожному кроці змінна **x** перезаписується, то неможливо зробити ще один перегляд прочитаних значень*

```
int main() {  
    int n, i;  
    double x, average = 0;  
    printf("n = "); scanf("%d", &n);  
    for (i = 0; i < n; i++) {  
        printf("a%d = ", i); scanf("%lf", &x);  
        average += x;  
    }  
    average /= n;  
    printf("Average = %f\n", average);  
    return 0;  
}
```



Тому в даному випадку для виконання поставленого завдання потрібно зберігати у пам'яті усі введені значення.

Для зберігання великої кількості однотипних даних використовують **масиви**.

---

**Масив** – набір однотипних значень, об'єднаних під одним ім'ям.

Значення називають **елементами** масиву.

Кожний елемент має свій унікальний номер, який називають **індексом**.

---

**Індекси** – це  
номери елементів  
масиву

*індекси*



0 1 2 3 4 5 6 7 8 9

-5	6	3	10	3	1	5	1	-3	5
----	---	---	----	---	---	---	---	----	---



*елементи*

*масив*



Масив, що складається з **10**-ти елементів

<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
<b>-5</b>	<b>6</b>	<b>3</b>	<b>10</b>	<b>3</b>	<b>1</b>	<b>5</b>	<b>1</b>	<b>-3</b>	<b>5</b>

Індекси у мові Сі починаються з **нуля**

Останній елемент має індекс **N-1** (в даному випадку – **9**)

---



# Синтаксис оголошення масиву:

тип ім'я масиву [ розмір ] ;



*кількість  
елементів*

Розмір масиву повинний бути  
вказаний явно у вигляді числа.  
Змінну використовувати тут неможна

---

Приклади оголошення масивів:

```
double arrDb1[20];  
int arrInt[50];  
char arrChar[33];
```

А так робити у мові Сі неможна:

```
int n;  
scanf("%d", &n);  
double arrDb1[n];
```

---

Для того, щоб записати значення в елемент масиву потрібно вказати ім'я масиву та індекс елемента у квадратних дужках:

```
double studGrades[26];  
studGrades[0] = 4;  
studGrades[1] = 5;  
studGrades[2] = 3;  
studGrades[3] = 4;  
studGrades[4] = 2;  
...
```

Якщо одразу на етапі написання програми є дані, які потрібно явно записати у масив, це можна зробити також більш зручним способом:

```
int yearOfBirth[26] = {1999,  
2000, 1999, 1998, 2001, 1998};
```

---

```
int yearOfBirth[26] = {1999, 2000,  
1999, 1998, 2001, 1998};
```

Те ж саме можна було б виконати і так, але тоді потрібно писати більше рядків програмного коду:

```
int yearOfBirth[26];  
yearOfBirth[0] = 1999;  
yearOfBirth[1] = 2000;  
yearOfBirth[2] = 1999;  
yearOfBirth[3] = 1998;  
yearOfBirth[4] = 2001;  
yearOfBirth[5] = 1998;
```

---

Для читання значень елементів масиву використовується така ж форма, як і при записі значення:

```
printf("First student : %d\n  
Second student : %d\n",  
       yearOfBirth[0],  
       yearOfBirth[1]);
```

---

При оголошенні масиву його елементи можуть містити непередбачувані значення!

```
int yearOfBirth[5];  
printf("First student : %d\n  
        Second student : %d\n",  
        yearOfBirth[0],  
        yearOfBirth[1]);
```

```
C:\Windows\system32\cmd.exe
```

```
First student : -858993460  
Second student : -858993460
```

Якщо при ініціалізації масиву у {...} вказано значення не всіх елементів, то інші автоматично отримують нульове значення.

```
int yearOfBirth[26] = {1999,  
2000, 1999, 1998, 2001, 1998};
```

Тут елементи з індексами 0, 1, 2, 3, 4, 5 отримують вказані значення, а елементи з 6-ого по 25-ий будуть містити нулі

---



Тобто найпростіший спосіб ініціалізувати масив нулями:

```
int myArray[100] = {0};
```

Елементу з індексом 0 буде присвоєно значення 0, оскільки не вказано значення для інших елементів, вони теж будуть обнулені.

---

```
int myArray[100] = {1};
```

А в даному прикладі елементу з індексом 0 буде присвоєно значення 1, оскільки не вказано значення для інших елементів, то вони матимуть нульові значення.

---

При зверненні до елемента масиву у квадратних дужках **можна використовувати змінні**.

Це дає можливість виконувати операції з елементами масиву у циклі.

```
int yearOfBirth[10];  
for (int i = 0; i < 10; i++) {  
    printf("yearOfBirth[%d] = ", i);  
    scanf("%d", &yearOfBirth[i]);  
}
```

---

```
int yearOfBirth[10];
for (int i = 0; i < 10; i++) {
    printf("yearOfBirth[%d] = ", i);
    scanf("%d", &yearOfBirth[i]);
}
```

C:\Windows\system32\cmd.exe

```
yearOfBirth[0] = 1998
yearOfBirth[1] = 1999
yearOfBirth[2] = 1999
yearOfBirth[3] = 2000
yearOfBirth[4] = 2001
yearOfBirth[5] = 1998
yearOfBirth[6] = 2000
yearOfBirth[7] = 2001
yearOfBirth[8] = 1998
yearOfBirth[9] = 1999
```

Повертаємось до завдання, яке було поставлене раніше.

**Приклад.** З клавіатури вводиться  $N$  чисел. Знайти кількість чисел, які менші за середнє арифметичне введених значень.



Середнє арифметичне = 2.6

Кількість шуканих чисел: 4

---

```
int main() {
    int n, i, count = 0;
    double arr[100], average = 0;
    printf("n = "); scanf("%d", &n);
    for (i = 0; i < n; i++) {
        printf("a%d = ", i);
        scanf("%lf", &arr[i]);
        average += arr[i];
    }
    average /= n;
    for (i = 0; i < n; i++) {
        if (arr[i] < average)
            count++;
    }
    printf("Average = %f\n", average);
    printf("Count = %d\n", count);
    return 0;
}
```

```
C:\Windows\system32\c
```

```
n = 10
```

```
a0 = -5
```

```
a1 = 6
```

```
a2 = 3
```

```
a3 = 10
```

```
a4 = 3
```

```
a5 = 1
```

```
a6 = 5
```

```
a7 = 1
```

```
a8 = -3
```

```
a9 = 5
```

```
Average = 2.600000
```

```
Count = 4
```



Написати програму яка виводить інформацію про кількість днів у кожному місяці.

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#define MONTHS 12

int main(){
system("chcp 1251");
int days[MONTHS] = { 31,28,31,30,31,30,31,31,30,31,30,31 };
for (int index = 0; index < MONTHS; index++)
printf("У %d місяці %d днів \n", index + 1, days[index]);
return 0;
}
```

---



Значення елементів масиву можна вводити в діалоговому режимі, в процесі роботи програми.




```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
int main(){
system("chcp 1251");
int arrey[10];

for (int i = 0; i < 10; i++){
printf("\nВведіть елемент масиву з індексом %d ", i);
scanf("%d ", &arrey[i]);
}
for (int i = 0; i < 10; i++)
printf("%4d ", arrey[i]);

return 0;
}
```

---



Для демонстрації роботи програми з масивами кожного разу доводиться вводити масив з клавіатури.

Зручніше його автоматично заповнити випадковими числами.

У програмуванні числа не випадкові, а **«псевдовипадкові»**.

---

Щоб генерувати псевдовипадкові числа треба:

1) підключити бібліотеку

```
#include<stdlib.h>
```

2) використати функцію

```
rand()
```

---

3) для генерування псевдовипадкового числа використовується функція **rand()**.

Вона генерує числа у діапазоні від 0 до 32767.

Щоб згенерувати числа з інтервалу [0; x):  
`rand() % x`

Щоб згенерувати числа з інтервалу [0; x]:  
`rand() % (x + 1)`

Щоб згенерувати числа з інтервалу [a; b]:  
`a + rand() % (b - a + 1)`

---

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>

int main(){
system("chcp 1251");
int arrey[10];

for (int i = 0; i < 10; i++)
arrey[i] = rand() %10;

for (int i = 0; i < 10; i++)
printf("%4d ", arrey[i]);
return 0;
}
```

```
Текущая кодовая страница: 1251
```

```
1 7 4 0 9 4 8 8 2 4
```

```
Текущая кодовая страница: 1251
```

```
1 7 4 0 9 4 8 8 2 4
```

# Щоб правильно використовувати функцію *rand* необхідно:

1) підключити бібліотеки

```
#include<stdlib.h>
```

```
#include<time.h>    або    #include <ctime>
```

2) на початку функції `main()` виконати код:

```
srand(time(0));
```



Вказується  
послідовність  
псевдовипадкових  
чисел

Кількість секунд,  
що пройшли від  
1 січня 1970 року

---

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <ctime>

int main(){
system("chcp 1251");
srand(time(0)); // автоматична рандомізація
int arrey[10];

for (int i = 0; i < 10; i++)
arrey[i] = rand() %10;

for (int i = 0; i < 10; i++)
printf("%4d ", arrey[i]);
return 0;
}
```

Текущая кодовая страница: 1251

6 0 0 0 3 2 6 0 1 5

Текущая кодовая страница: 1251

8 2 4 5 9 4 3 9 7 7

Ввести з клавіатури масив з 10 елементів, помножити всі елементи на 2 і вивести отриманий масив на екран.



Дан одновимірний цілочисельний масив  $A$ , що складається з  $N$  елементів,  $N$ -задане натуральне число. Знайти суму всіх елементів масиву, якщо серед них є хоча б один, більший одиниці, і добуток- в іншому випадку.

