

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Інститут телекомунікації, радіоелектроніки та електронної техніки

кафедра «Радіоелектронні пристрої та системи»  
З дисципліни «Програмування частина 2»  
Лабораторна робота №15-16  
«Дослідження використання одновимірних та багатовимірних масивів»

Мета роботи: Навчитися використовувати одновимірні та багатовимірні масиви у процесі програмування для обробки великої сукупності значень.

Підготував  
ст.групи АП-11  
Василюк Ростислав

Прийняв:  
Чайковський І.Б.

Львів 2024р

## Теоретичні відомості:

В усіх програмах, що розглядалися у лабораторних роботах, оброблялися поодинокі значення. На практиці часто виникає потреба обробити єдиним алгоритмом велику сукупність однорідних значень. В математиці такі сукупності мають позначення на зразок  $x_1, x_2, \dots, x_n$ .

Для доступу до елементу масиву слугує операція індексування, яка позначається квадратними дужками: конструкція `ім'я_масиву[номер_елементу]` означає змінну - елемент, що стоїть в даному масиві під даним номером. Номер елементу в масиві називають також індексом.

```
1.int m[10], k=3;
2.m[0]=1;
3.m[k]=8;
4.++ k;
5.m[k]=8;
6.m[( k + 2)%3+ 1]=17;
7.m[k+3]=m [0]+ m[k];
8.scanf("% d",&m[k+1]);
9.printf("%d\ n",m[k]);
```

В першому рядку оголошується масив `m` з 10 елементів та допоміжна змінна `k`, яка одразу отримує початкове значення 3. В рядку 2 показано, як присвоїти значення елементу масиву, номер якого заздалегідь відомий: в якості індексу використано константу, число 0. Оскільки нумерація елементів починається з 0, то даний оператор означає, що значення присвоюється першому елементу масиву. Рядок 3 ілюструє, що індекс може бути не константою, а значенням змінної. Оскільки в даний момент змінна `k` має значення 3, даний оператор означає, що значення 8 присвоюється у четвертий від початку (а не третій!) елемент масиву. Оператор в рядку 4 збільшує значення змінної `k` на 1, отже, воно тепер дорівнює 4. Тому, хоча оператор в рядку 5 повністю співпадає за написанням з оператором в рядку 3, тепер вираз в лівій частині присвоювання означає вже не четвертий, а п'ятий від початку елемент масиву. Рядок 6 є прикладом того, що в якості індексу може використовуватися не лише значення змінної, але і складний вираз.

Підставивши поточне значення змінної `k`, маємо, що значення 17 буде присвоєно елементові з індексом 1, тобто другому елементу масиву. В рядку 7 показано, що звертання до елементів одного й того самого масиву може здійснюватися і в лівій, і в правій частинах присвоювання. В перший 3 (з індексом 0) елемент раніше було занесене значення 1, поточне значення змінної `k` дорівнює 4, а елементу з індексом 4 було присвоєно значення 8. Отже, елемент з індексом 7 (восьмий від початку) отримає значення 9. Значення елементів масиву можна вводити з клавіатури так само, як і значення звичайних змінних, за допомогою функції `scanf`, що показано в рядку 8. Як і завжди, перед іменем змінної, в яку треба розмістити введене значення, ставиться знак `&` - амперсанд. З рядка 9 видно, що значення

елементів масиву можна передавати до функцій в якості аргументів, в тому числі - друкувати на екран.

Хід роботи:

1. Ознайомитися з теоретичними відомостями.
  2. Здійснити виконання прикладів, представлених у теоретичних відомостях, після чого представити скріни їх коду та результати виконання у звіті.
- Ці функції оголошені в заголовному файлі . Застосування бібліотечних функцій обробки рядків ілюструється наступним прикладом:



```
1 #include <stdio.h>
2 #include <string.h>
3
4 #define MAX 100
5 #define LEN 80
6
7 char text[MAX][LEN];
8
9 int main(void) {
10     int t, i, j;
11
12     printf("Для виходу введіть порожній рядок\n");
13     for (t = 0; t < MAX; ++t) {
14         printf("t: ", t);
15         getc(text[t]);
16         if (!text[t])
17             break;
18     }
19
20     for (i = 0; i < t; ++i) {
21         for (j = 0; text[i][j]; ++j)
22             putchar(text[i][j]);
23     }
24     return 0;
25 }
```

Input: 0123456

Output: 0\n1\n2\n3\n4\n5\n6\n

Нижче представлено програму для обчислень значення елементів одновимірного масиву цілого типу з індексами від 0 до 99.

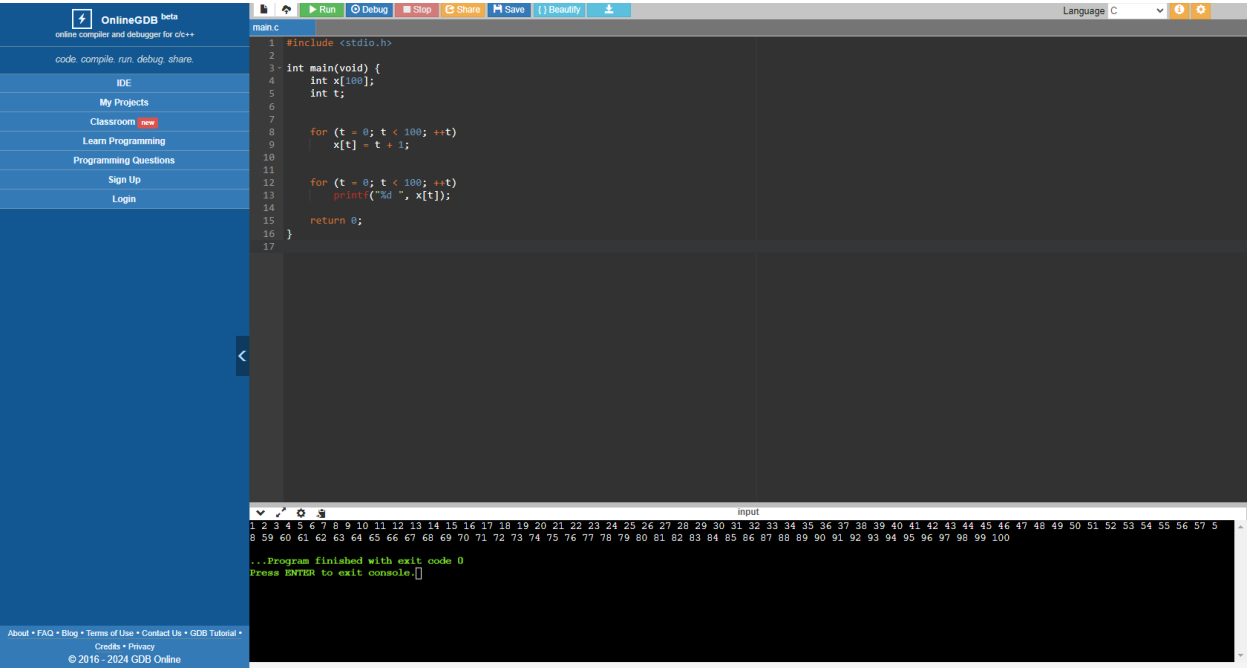


```
1 #include <stdio.h>
2
3 int main(void) {
4     int t, i, num[100];
5
6     for (t = 0; t < 5; ++t) {
7         for (i = 0; i < 4; ++i) {
8             num[t][i] = (t * 4) + i + 1;
9         }
10    }
11
12    for (t = 0; t < 5; ++t) {
13        for (i = 0; i < 4; ++i) {
14            printf("%3d ", num[t][i]);
15        }
16        printf("\n");
17    }
18    return 0;
19 }
```

Output: 1 2 3 4\n5 6 7 8\n9 10 11 12

...Program finished with exit code 0  
Press ENTER to exit console.

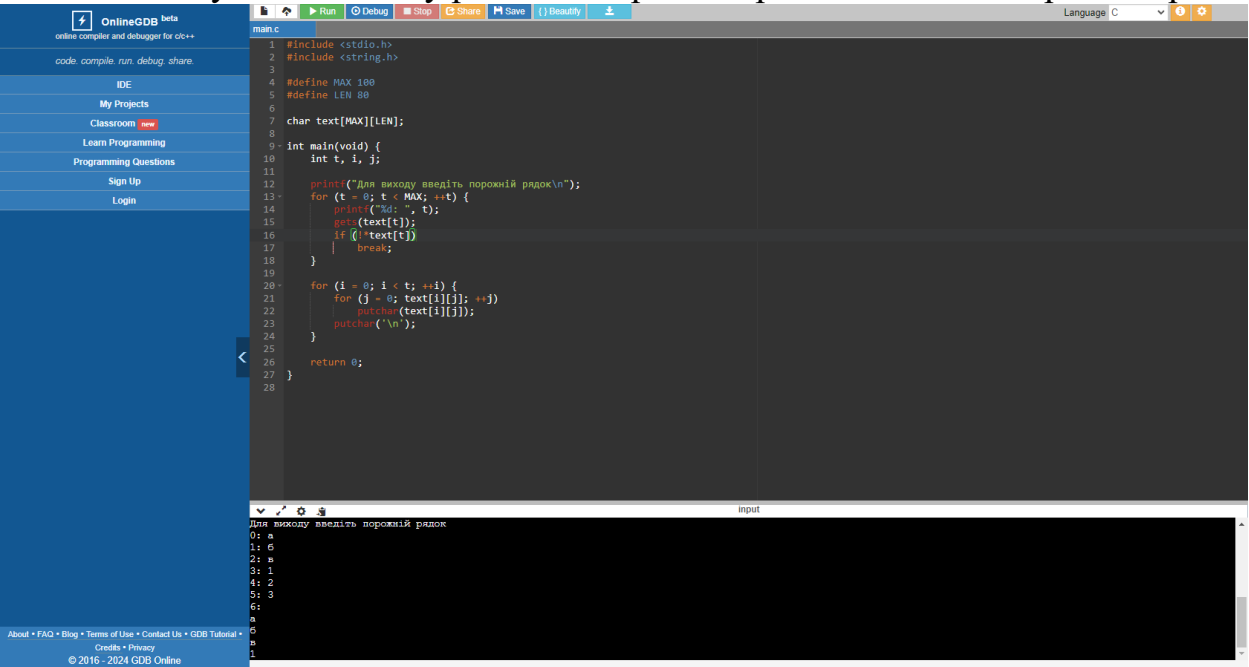
У наступному прикладі елементам двовимірного масиву присвоюються числа від 1 до 12 і значення елементів виводяться на екран порядково:



```
1 #include <stdio.h>
2
3 int main(void) {
4     int x[100];
5     int t;
6
7     for (t = 0; t < 100; ++t)
8         x[t] = t + 1;
9
10    for (t = 0; t < 100; ++t)
11        printf("%d ", x[t]);
12
13    return 0;
14 }
```

...Program finished with exit code 0  
Press ENTER to exit console.

З цих двох форм запису кращою є перша. Для кращого розуміння властивостей масиву рядків розглянемо наступну коротку програму, в якій на основі застосування масиву рядків створений простий текстовий редактор:



```
1 #include <stdio.h>
2 #include <string.h>
3
4 #define MAX 100
5 #define LEN 80
6
7 char text[MAX][LEN];
8
9 int main(void) {
10     int t, i, j;
11
12     printf("Для виходу введіть порожній рядок\n");
13     for (t = 0; t < MAX; ++t) {
14         printf("%d: ", t);
15         getc(text[t]);
16         if (getc(text[t])
17             | break;
18     }
19
20     for (i = 0; i < t; ++i) {
21         for (j = 0; text[i][j]; ++j)
22             putchar(text[i][j]);
23         putchar('\n');
24     }
25
26     return 0;
27 }
```

Для виходу введіть порожній рядок  
0:   
1:   
2:   
3:   
4:   
5:   
6:   
a  
s

3. Напишіть програму, яка заповнює масив довільними цілими числами, введеними з клавіатури (розмір масиву становить 10), обчислює суму всіх елементів та виводить на екран сам масив і значення суми його елементів.



```
1 #include <stdio.h>
2
3 int main() {
4     int array[10];
5     int sum = 0;
6
7     printf("Введіть 10 цілих чисел:\n");
8     for (int i = 0; i < 10; ++i) {
9         scanf("%d", &array[i]);
10    }
11
12    for (int i = 0; i < 10; ++i) {
13        sum += array[i];
14    }
15
16    printf("Масив:\n");
17    for (int i = 0; i < 10; ++i) {
18        printf("%d ", array[i]);
19    }
20    printf("\nСума елементів масиву: %d\n", sum);
21    return 0;
22 }
```

Введіть 10 цілих чисел:

2  
3  
5  
6  
7  
8  
12  
34  
25  
68

Масив:  
2 3 5 6 7 8 12 34 25 68

Сума елементів масиву: 170

...Program finished with exit code 0  
Press ENTER to exit console.

Контрольні запитання:

1. Дайте визначення поняття масив в мові С.

Це структура даних, яка представляє собою колекцію однотипних елементів, що зберігаються у пам'яті підрядково, тобто в одному рядку. Кожен елемент масиву має свій унікальний індекс, за допомогою якого його можна звертатися.

2. Назвіть види масивів.

Одновимірні масиви: масив, у якого один індекс.

Багатовимірні масиви: масив, у якого два або більше індексів.

Статичні масиви: масив, розмір якого відомий на етапі компіляції.

Динамічні масиви: масив, розмір якого визначається під час виконання програми.

3. Назвіть перевагу використання багатовимірних масивів.

Багатовимірні масиви дозволяють представляти більш складні структури даних, такі як матриці або зображення. Вони дозволяють зручно організовувати дані в багатовимірних структурах, що спрощує їх обробку та доступ до них.

4. Для чого у масивах використовується матриця?

Це багатовимірний масив, який представляється у вигляді таблиці з рядками та стовпцями. Вона використовується для зберігання та обробки даних, які мають двовимірну структуру, наприклад, зображення, табличні дані тощо.

5. Яка загальна форма ініціалізації масиву?

Загальна форма ініціалізації масиву виглядає так: тип\_даних

ім'я\_масиву[розмір] = {елемент\_1, елемент\_2, ..., елемент\_n};. Тут тип\_даних

- це тип даних елементів масиву, ім'я\_масиву - назва масиву, розмір -

кількість елементів у масиві, а {елемент\_1, елемент\_2, ..., елемент\_n} - це

список значень, з яких складається масив.