

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
«НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Інститут телекомунікацій, радіоелектроніки та електронної техніки
Кафедра «Радіоелектронні пристрої та системи»



Звіт з лабораторної роботи № 15-16
«Програмування, частина 2»

Підготував:
ст. гр. АП-11
Заброварний Олег
Перевірив:
Асистент каф РЕПС
Чайковський І.Б.

Львів 2024

Тема: «Дослідження використання одновимірних та багатовимірних масивів»

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

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 підтримки обробки таких сукупностей в мові C існує поняття масиву. Масив - це сукупність даних одного типу, що об'єднані спільним ім'ям.

```
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 (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. Одразу ж при оголошенні масиву можна присвоювати значення його елементам, або, як кажуть, ініціалізувати масив.

Функції для обробки рядків

Функція	Виконувана дія
<i>strcpy(s1,s2)</i>	Копіювання <i>s2</i> в <i>s1</i>
<i>strcat(s1,s2)</i>	Конкатенація (приєднання) <i>s2</i> в кінець <i>s1</i>
<i>strlen(s1)</i>	Повертає довжину рядка <i>s1</i>
<i>strcmp(s1,s2)</i>	Повертає 0, якщо <i>s1</i> і <i>s2</i> збігаються, негативне значення, якщо <i>s1</i> < <i>s2</i> і позитивне значення, якщо <i>s1</i> > <i>s2</i>
<i>strchr(s1,ch)</i>	Повертає вказівник на перше входження символу <i>ch</i> в рядок <i>s1</i>
<i>strstr(s1,s2)</i>	Повертає вказівник на перше входження рядка <i>s2</i> в рядок <i>s1</i>

Графічне представлення двовимірного масиву

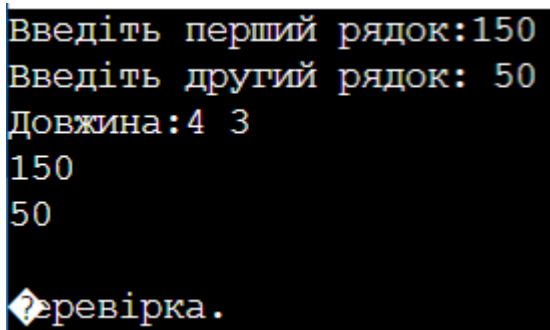
Масив <i>a</i>	Стовбець 0	Стовбець 1	Стовбець 2
Рядок 0	<i>a</i> [0][0]	<i>a</i> [0][1]	<i>a</i> [0][2]
Рядок 1	<i>a</i> [1][0]	<i>a</i> [1][1]	<i>a</i> [1][2]
Рядок 2	<i>a</i> [2][0]	<i>a</i> [2][1]	<i>a</i> [2][2]
Рядок 3	<i>a</i> [3][0]	<i>a</i> [3][1]	<i>a</i> [3][2]

Завдання 1

```

#include <stdio.h>
#include <string.h>
int main(void) {
    char s1[80], s2[80];
    printf("Введіть перший рядок:");
    fgets(s1, sizeof(s1), stdin);
    printf("Введіть другий рядок: ");
    fgets(s2, sizeof(s2), stdin);
    printf("Довжина:%zu %zu\n", strlen(s1), strlen(s2));
    if (!strcmp(s1, s2))
        printf("Рядки рівні:\n");
    strcat(s1, s2);
    printf("%s\n", s1);
    strcpy(s1, "Перевірка.\n");
    printf("%s", s1);
    return 0;
}

```



```

Введіть перший рядок:150
Введіть другий рядок: 50
Довжина:4 3
150
50
Перевірка.

```

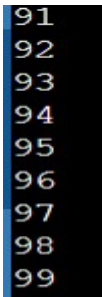
Завдання 2

```

#include <stdio.h>
int main(void) {
    int x[100]; //оголошення масиву цілого типу з 100 чисел

```

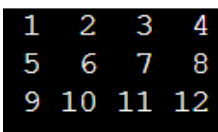
```
int t;  
for(t=0;t<100;++t) x[t]=t; //присвоєння значення від 0 до 99  
for(t=0;t<100;++t) printf("%d\n",x[t]);}
```



```
91  
92  
93  
94  
95  
96  
97  
98  
99
```

Завдання 3

```
#include<stdio.h>  
  
int main(void){  
    int t,i, num[3][4];  
    for(t=0;t<3;++t)  
        for(i=0;i<4;++i)  
            num[t][i]=(t*4)+i+1;  
    //вивід на екран  
    for(t=0;t<3;++t){  
        for(i=0;i<4;++i)  
            printf("%3d",num[t][i]);  
        printf("\n");  
    }  
    return 0;  
}
```



```
1  2  3  4  
5  6  7  8  
9 10 11 12
```

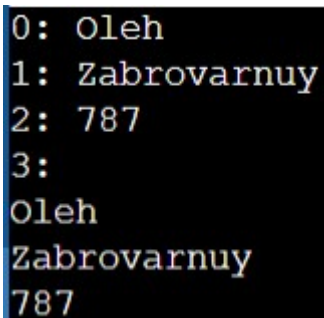
Завдання 4

```

#include <stdio.h>

#define MAX 100
#define LEN 80
char text[MAX][LEN];
int main(void) {
    int t, i, j;
    printf("Для виходу введіть пустий рядок.\n");
    for (t = 0; t < MAX; t++) {
        printf("%d: ", t);
        gets(text[t]);
        if (!*text[t]) break; } // вихід при пустому рядку
    for (i = 0; i < t; i++) {
        for (j = 0; text[i][j]; j++)
            putchar(text[i][j]);
        putchar('\n'); }
    return 0;
}

```



```

0: Oleh
1: Zabrovarnuy
2: 787
3:
Oleh
Zabrovarnuy
787

```

Завдання 5

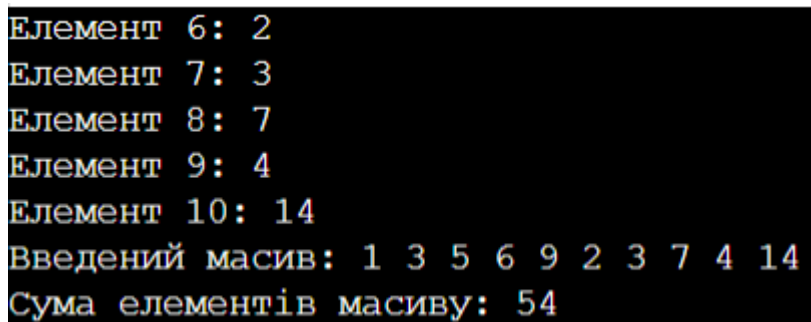
```

#include <stdio.h>

#define SIZE 10
int main() {

```

```
int arr[SIZE];
int sum = 0;
// Заповнення масиву
printf("Введіть %d цілих чисел:\n", SIZE);
for (int i = 0; i < SIZE; i++) {
    printf("Елемент %d: ", i + 1);
    scanf("%d", &arr[i]);
    sum += arr[i]; // Додавання поточного елементу до суми
}
// Виведення масиву
printf("Введений масив: ");
for (int i = 0; i < SIZE; i++) {
    printf("%d ", arr[i]);
}
// Виведення суми елементів масиву
printf("\nСума елементів масиву: %d\n", sum);
return 0;}
```



Елемент 6: 2
Елемент 7: 3
Елемент 8: 7
Елемент 9: 4
Елемент 10: 14
Введений масив: 1 3 5 6 9 2 3 7 4 14
Сума елементів масиву: 54

Висновок: На даній лабораторній роботі я навчитися використовувати одновимірні та багатовимірні масиви у процесі програмування для обробки великої сукупності значень.