

Лабораторна робота № 8

МАСИВИ.

Мета роботи:

Вивчити поняття і застосування масивів.

Обладнання та програмне забезпечення:

- IBM сумісна персональна обчислювальна машина;
- онлайн компілятор мови програмування Cі, доступний за посиланням <https://repl.it/languages/c>

Завдання до роботи:

- Написати програму на мові програмування Cі, котра використовує наступні дії над одновимірними і двовимірними масивами:
- пошук довільного елементу;
 - пошук мінімального та максимального елементу;
 - знаходження суми елементів.

Теоретичні відомості

Масиви

Масив - це сукупність змінних одного типу, до яких звертаються за допомогою загального імені. Доступ до окремого елементу масиву може здійснюватися за допомогою індексу. В Cі всі масиви складаються з сусідніх ділянок пам'яті. Найменша адреса відповідає першому елементу. Найбільша адреса відповідає останньому елементу. Масиви можуть мати одну або кілька розмірностей. Масиви досить тісно пов'язані з вказівниками.

Оголошення масиву:

`специфікатор_типу імя_змінної [розмір];`

У всіх масивах перший елемент має індекс 0. Тому, якщо написати:

```
int a [10];
```

то буде оголошений масив з цілих чисел з 10 елементів, причому ці елементи адресуються індексом від 0 до 9. Наступна програма завантажує цілочисельний масив числами від 0 до 9 і виводить його:

```
#include <stdio.h>
int main(void)
{
    int a[10]; // резервує місце для 10-ти цілочисельних елементів
    int i;
    for(i=0; i<10; ++i) a[i] = i;
    for(t=0; i<10; ++i) printf("%d ", a[i]);
    return 0;
}
```

В Сі відсутня перевірка меж масивів. Можна вийти за один кінець масиву і записати значення в будь-яку змінну, що не відноситься до масиву, або навіть в код програми.

Одновимірні масиви це насправді списки інформації одного типу. Наприклад, таблиця показує, як масив `a` розташовується в пам'яті, якщо перший елемент має адресу 105:

Елемент	<code>a[0]</code>	<code>a[1]</code>	<code>a[2]</code>	<code>a[3]</code>	<code>a[4]</code>	<code>a[5]</code>	<code>a[6]</code>	<code>a[7]</code>	<code>a[8]</code>	<code>a[9]</code>
Адреса	105	106	107	108	109	110	111	112	113	114

Можна створити вказівник на перший елемент масиву, вказавши ім'я масиву без індексу. Нехай є

```
int mas[10];
```

Наступний фрагмент коду присвоює змінній `p` адресу першого елемента `mas`:

```
int *p;  
int mas[10];  
p = mas;
```

Можна також отримати адресу першого елемента масиву за допомогою оператора `&`. Наприклад, `mas` і `&mas[10]` призводять до однакового результату.

При передачі одновимірних масивів у функції слід викликати функцію з ім'ям масиву без індексу. В результаті цього передається адреса першого елемента масиву. В Сі неможливо передати весь масив як аргумент. Замість цього автоматично передається вказівник. Наступний приклад передає адресу `mas` в `func1()`;

```
int main (void)  
{  
    int mas[10];  
    func1(mas);  
    ...  
}
```

Якщо функція отримує одновимірний масив, то можна оголосити формальний параметр як вказівник, як масив з фіксованою довжиною або як безрозмірний масив. Наприклад, для передачі `mas` в функцію `func1()` можна оголосити `func1()` одним із таких способів:

```
void fun1 (int * a) // вказівник  
{  
    ...  
}
```

або

```
void fun1 (int a [10]) //масив з фіксованою довжиною
{
    ...
}
```

або

```
void fun1 (int a []) // безрозмірний масив
{
    ...
}
```

Сі дозволяє створювати багатовимірні масиви. Найпростішим видом багатовимірного масиву є двовимірний масив. Двовимірний масив - це масив одновимірних масивів. Двовимірний масив оголошується наступним чином:

```
специфікатор_типу імя_змінної [розмір 2-го виміру] [розмір 1-го виміру];
```

Приклад:

```
#include <stdio.h>
int main(void)
{
    int i,j, matr[3][4];
    for(j=0; j<3; ++j)
        for (i=0; i<4; ++i)
            matr[j][i] = (j*10)+i;
    // вивід чисел
    for(j=0; j<3; ++j)
    {
        for (i=0; i<4; ++i)
            printf("%2d ",matr[j][i]);
        printf ("\n");
    }
    return 0;
}
```

Генерація випадкового числа:

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

int main() {
    srand(time(NULL)); // Initialization, should only be called once
    int r = rand() % 20; // Random int between 0 and 19
    printf("%d\n",r);
}
```

ПОРЯДОК ВИКОНАННЯ РОБОТИ:

1. Опрацювати і засвоїти матеріал, наведений в теоретичних відомостях.
2. Написати функцію, яка повертає масив з 10 випадкових чисел.
3. Написати функції, що повертають вказівник на довільний елемент одновимірного масиву з 10 елементів та двовимірного масиву зі 100 елементів (повернути NULL, якщо елемент відсутній).
4. Написати функцію, котра знаходить максимальний і мінімальний елементи масиву та їхню суму.
5. В звіті навести копії екранів та написаний код.
6. Зробити висновки.