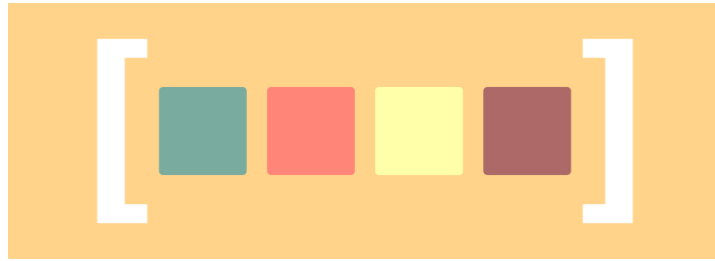


C++ масиви

Андрій Янковський

2020



Масив

Це набір фіксованої кількості
однотипних елементів, що мають
порядковий номер і спільне ім'я

Ось так виглядає пустий масив на **5**
елементів:

```
int mas[5];
```



`int` – каже нам, що це масив цілих чисел, `mas` – його назва, а `[5]` - виділяють пам'ять під 5 елементів масиву.

А як записати якесь значення в перший елемент масиву?

```
mas[0] = 0;
```

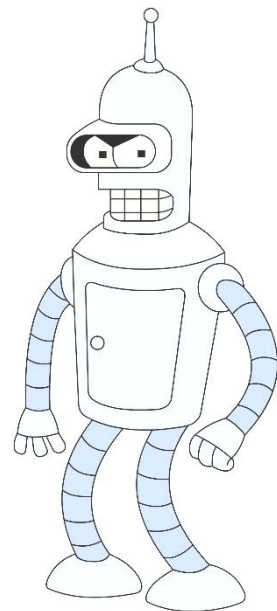
А в інші елементи?

```
mas[1] = 7;
```

```
mas[2] = 5;
```

```
mas[3] = 3;
```

```
mas[4] = 1;
```



А в п'яту комірку як записати?

А у нас вже всі п'ять комірок зайняті 0, 1, 2, 3, 4, а 5-ї **не має**. Річ у тім, що в розмірі масива ми вказуємо кількість елементів, але їх нумерація іде не з 1, а з 0 – саме тому останній елемент масиву на одиничку менший за його розмір!

Виведемо масив **mas** на екран:

```
cout << mas[0] << ", ";
```

```
cout << mas[1] << ", ";
```

```
cout << mas[2] << ", ";
```

```
cout << mas[3] << ", ";
```

```
cout << mas[4]);
```

0, 7, 5, 3, 1

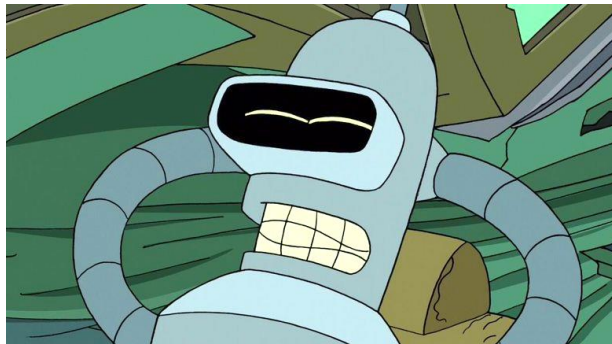
А є швидший спосіб для виводу?

Звичайно!

```
for ( int i = 0; i < 5; i++ ){  
    cout << mas[i] << " ";  
}
```

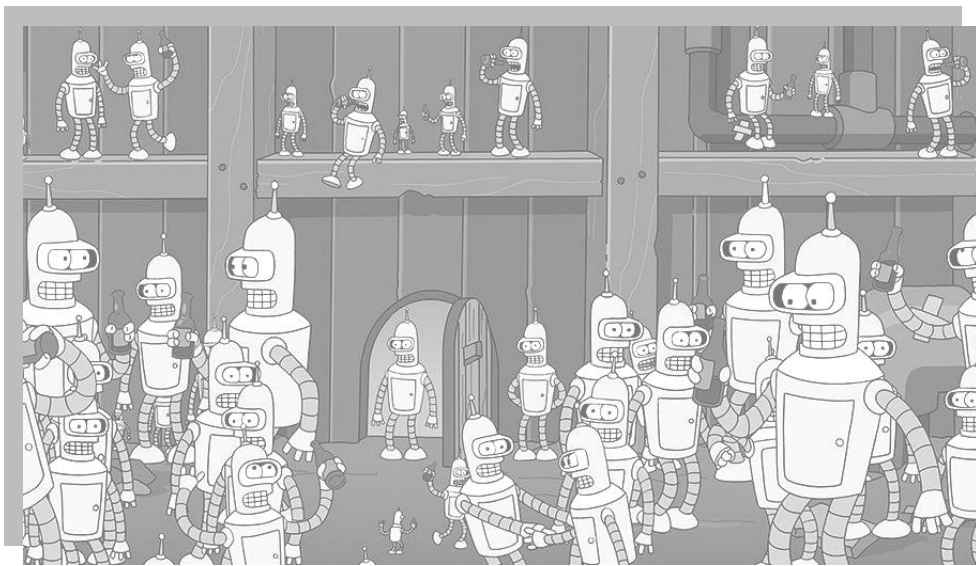
Зміна **i** буде змінюватися від 0 (включно) до 5 (не включно), тобто **0, 1, 2, 3, 4** і так ми зможемо кожен раз брати наступний елемент масиву та виводити його на екран:

0 7 5 3 1



Так, ми ще можемо записати шість цифр в шість змінних:

```
int a1 = 1;    int a2 = 3;  
int a3 = 5;    int a4 = 7;  
int a5 = 9;    int a6 = 11;
```



Але навіщо писати так багато, якщо є масив і ось така можливість:

```
int mas[6] = { 1, 3, 5, 7, 9, 11 };
```

Не можна в один масив записувати змінні: і типу `int`, і типу `double`, і типу `string` – ви маєте обрати щось одне!

```
string array[3] = { "A1", "B1", "C1" };
```

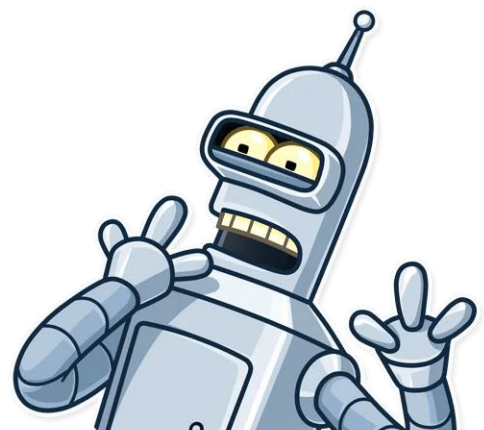
```
string array[3] = { 1, 3.14, "C1" };
```

```
int array[3] = { 1, 3.14, "C1" };
```

```
int array[3] = { 6, 3, 9 };
```

А яка найбільша проблема при роботі з масивами?

Те, що люди часто в циклі виходять за межі масиву і програма припиняє роботу.



А можна заповнити масив
автоматично?

Та без проблем!

```
for ( int i = 0; i < 6; i++ ){  
    mas[i] = i;  
}
```

0, 1, 2, 3, 4, 5

```
for ( int i = 0; i < 6; i++ ){  
    mas[i] = i + 1;  
}
```

1, 2, 3, 4, 5, 6


```
for ( int i = 6; i > 0; i-- ){  
    mas[i] = i;  
}
```

6, 5, 4, 3, 2, 1

```
for ( int i = 0; i < 6; i++ ){  
    mas[i] = (i + 1) * (i + 1);  
}
```

1, 4, 9, 16, 25, 36

```
for ( int i = 10; i <= 60; i = i + 10 ){  
    mas[i] = i;  
}
```

10, 20, 30, 40, 50, 60



А ми можемо заповнити масив випадковими числами?

```
#include <iostream>
#include <time.h>
using namespace std;
```

```
int main()
{
    srand(time(0));
    int mas[10];
    for ( int i = 0; i < 10; i++ ){
        mas[i] = rand() % 10;
    }

    system("pause");
    return 0;
}
```

Так ми заповнимо масив випадковими числами від 1 до 9.

Число 0 – може зустрічатися, а 10 – ні, бо генератор генерує числа на одиницю менші від вказаного числа;

А давайте додамо перший елемент першого масиву з першим елементом другого масиву, а другий з другим... і запишимо результат в третій масив

```
int mas1[5] = {1, 2, 3, 4, 5};
```

```
int mas2[5] = {1, 1, 1, 1, 1};
```

```
int mas3[5];
```

```
for ( int i = 0; i < 5; i++ ){  
    mas3[i] = mas1[i] + mas2[i];  
}
```

А тепер давайте спробуємо об'єднати усі значення двох масивів в третьому

```
int mas1[5] = { 1, 2, 3, 4, 5 };
```

```
int mas2[5] = { 1, 1, 1, 1, 1 };
```

```
int mas3[10];
```

```
int p = 0;
```

```
for ( int i = 0; i < 5; i++ ){
```

```
    mas3[p] = mas1[i];
```

```
    p++;
```

```
}
```

```
for ( int i = 0; i < 5; i++ ){
```

```
    mas3[p] = mas2[i];
```

```
    p++;
```

```
}
```

Бендер бажає тобі успіху!

