

C++ bir-ölçülü massivlər (1D arrays)

- Massivlər eyni-tipli bir çox sayda dəyişənləri saxlamaq üçün istifadə olunur.
- Massivi elan etmək üçün ilk olaraq massivin tipi, sonra isə adı və kvadrat mötərizədə (square brackets) ölçüsü (massiv üçün yaddaşda ayrılan yerlərin sayı), yəni elementlərinin sayı göstərilir.

```
#include <iostream>
using namespace std;
int main()
{
    int a[5];
}
```

Bu proqramda **int** data tipində **5** ölçülü **a** massivi təyin etmişik. Bu o deməkdir ki, massivdə **5** sayda **int** data tipində element saxlayacağıq.

```
#include <iostream>
using namespace std;
int main()
{
    double a[10];
}
```

Bu proqramda **double** data tipində **10** ölçülü **a** massivi təyin etmişik. Bu o deməkdir ki, massivdə **10** sayda **double** data tipində element saxlayacağıq.

```
#include <iostream>
using namespace std;
int main()
{
    string name[3];
}
```

Bu proqramda **string** tipində **3** ölçülü **name** massivi təyin etmişik. Bu o deməkdir ki, massivdə **3** sayda **string** tipində element saxlayacağıq.

- Yuxarıda massivləri təyin edilməsini öyrəndik, indi isə təyin edilən massivə elementlərin daxil edilməsinə baxaq.
- **Qeyd:** massivin elementləri soldan-sağa doğru **0**-dan başlayaraq nömrələnir.

```
#include <iostream>
using namespace std;
int main()
{
    string cars[3];
    cars[0] = "Volvo";
    cars[1] = "BMW";
    cars[2] = "Ford";
}
```

Bu proqramda **string** tipində **3** ölçülü **cars** massivi təyin etmişik və təyin etdikdən sonra hər sətirdə elementlərə dəyərlər vermişik. Massivin elementləri sıfırdan başlayaraq nömrələndiyi üçün, 0-cı indeks **Volvo**, 1-ci indeks **BMW**, 2-ci indeks isə **Ford** dəyərinə bərabərdir.

```
#include <iostream>
using namespace std;
int main()
{
    string cars[3] = {"Volvo", "BMW", "Ford"};
}
```

Burada təyin etdiyimiz massiv bir əvvəlki ilə eyni semantikanı ifadə edir. Sadəcə burada dəyərləri birbaşa massivi təyin edərkən yazmışıq.

```
#include <iostream>
using namespace std;
int main()
{
    int x[5] = {10, -5, 6, 2, 8};
}
```

Burada 5 ölçülü x massivi təyin etmişik və həmçinin dəyərlərini də yazmışıq. Massivin 0-cı indeksi 10, 1-ci indeksi -5, 2-ci indeksi 6, 3-cü indeksi 2 və 5-ci indeksi 8 dəyərinə bərabər olacaq.

- Massivin elementlərini çıxışa vermək üçün massivin adını və kvadrat mötərizədə çıxışa vermək istədiyimiz elementin indeksini yazmalıyıq.

```
#include <iostream>
using namespace std;
int main()
{
    string car[3] = {"BMW", "Mazda", "Volvo"};
    cout<<car[1];
}
```

Mazda

- **Qeyd:** massivin indeksini yazarkən verilən ölçüdən kənara çıxsaq həmin indeks üçün yaddaşda yer ayrılmadığına görə proqramda icra xətası (**error**) baş verəcək,
- Massivin hər hansı elementinin dəyərini dəyişmək üçün mənimsətmə operatorundan istifadə edə bilərik.

```
#include <iostream>
using namespace std;
int main()
{
    string cars[3] = {"Volvo", "BMW", "Ford"};
    cout<<cars[0]<<endl;
    cars[0] = "Opel";
    cout << cars[0];
}
```

Volvo
Opel

- Massivin bütün indekslərindəki dəyərləri çıxışa vermək üçün **for** dövründən və **cout<<** əmrindən istifadə etmək lazımdır.

```
#include <iostream>
using namespace std;
int main()
{
    int a[8] = {10, 5, 1, 2, 4, 6, 8, 7};
    for(int i = 0; i < 8; i++){
        cout<<a[i]<<" ";
    }
}
```

10 5 1 2 4 6 8 7

for-each dövrü

- **for-each** dövrü massiv elementlərinə indeks istifadə etmədən daha sürətli müraciət etmək üçün istifadə olunur.
- Bu dövrədən istifadə edərkən artma/azalma və dövrün şərt hissəsi istifadə olunmur.
- C dilində bu dövrə yoxdur, lakin C++ dilində mövcuddur. Bu dövrə C++11 versiyası və ya daha sonrakı versiyalar dəstəkləyir.
- **for-each** dövründə istifadə olunan açar söz **“for”** - dur.

```
#include <iostream>
using namespace std;
int main()
{
    int a[6] = {5, 1, 4, 2, 8, 6};
    for(int x : a){
        cout<<x<<endl;
    }
}
```

```
5
1
4
2
8
6
```