

در برنامه نویسی، یکی از مسائل تکراری ناشی از این است که داده های متعدد از یک نوع در یک دسته داشته باشیم. در نظر بگیرید که در حال بررسی ۱۰۰ نفر هستید و باید معدل آن ها را ذخیره کنید. برای حل این مشکل در C++ می توانید یک آرایه عدد صحیح با داشتن ۱۰۰ عنصر ایجاد کنید.

یک آرایه مجموعه ای از داده هایی است که تعدادی مقادیر هم نوع را نگه می دارد. مثلا:

```
int m[۱۰۰];
```

در اینجا، آرایه m می تواند حداکثر ۱۰۰ عنصر از نوع عدد صحیح را نگه دارد.

اندازه و نوع آرایه ها نمی تواند پس از تعریف تغییر کند.

چگونه می توان یک آرایه در C++ تعریف کرد؟

```
dataType arrayName[arraySize];
```

برای مثال

```
float m [۵];
```

در اینجا، ما یک آرایه m ، از نوع `float` و اندازه ۵ تعریف کردیم. یعنی می تواند ۵ عدد `float` را نگه دارد.

عناصر یک آرایه و نحوه دسترسی به آن ها

با استفاده از شاخص ها می توانید به عناصر آرایه دسترسی پیدا کنید.

فرض کنید شما آرایه m را در بالا تعریف کردید. عنصر اول $m[۰]$ است، عنصر دوم $m[۱]$ و به همین ترتیب.

نکات کلیدی:

- خانه شروع آرایه ها ۰ است نه ۱. در این مثال $m[۰]$ اولین عنصر است.
- اگر اندازه یک آرایه n باشد، برای دسترسی به آخرین عنصر، از $(n-۱)$ استفاده می شود. در این مثال، $m[۴]$ آخرین عنصر است.

چگونه می توان یک آرایه در C++ را مقدار دهی کرد؟

می توان به هنگام تعریف آرایه آن را مقدار دهی کرد. مثلا،

```
int m [۵] = {۱۹, ۱۰, ۸, ۱۷, ۹};
```

روش دیگر برای مقداردهی به آرایه در هنگام تعریف:

```
int m [] = {۱۹, ۱۰, ۸, ۱۷, ۹};
```

در این جا:

m [0] is equal to 19

m [1] is equal to 10

m [2] is equal to 8

m [3] is equal to 17

m [4] is equal to 9

چگونه می توان عناصر آرایه را وارد و چاپ کرد؟

```
۱. int m [5] = {19, 10, 8, 17, 9}
۲. // change 4th element to 9
۳. m [3] = 9;
۴. // take input from the user and insert in third element
۵. cin >> m [2];
۶. // take input from the user and insert in (i+1)th element
۷. cin >> m [i];
۸. // print first element of the array
۹. cout << m [0];
۱۰. // print ith element of the array
۱۱. cout >> m [i-1];
```

مثال: آرایه در C++

برنامه زیر ذخیره و محاسبه مجموع ۵ عدد وارد شده توسط کاربر را با استفاده از آرایه ها نشان می دهد.

```
۱. #include <iostream>
۲. using namespace std;
۳. int main()
۴. {
۵. int numbers[5], sum = 0;
۶. cout << "Enter 5 numbers: ";
۷. for (int i = 0; i < 5; ++i)
۸. {
۹. cin >> numbers[i];
۱۰. sum += numbers[i];
۱۱. }
۱۲. cout << "Sum = " << sum << endl;
۱۳. return 0;
۱۴. }
```

خروجی

```
Enter 5 numbers: 3
```

```
۴
```

```
۵
```

```
۴
```

```
۲
```

```
Sum = 18
```

مواردی که باید در هنگام کار با آرایه ها در C++ به یاد داشته باشید

فرض کنید یک آرایه از ۱۰ عنصر را تعریف کردید.

```
int testArray[۱۰];
```

می توانید از اعضای آرایه از

```
testArray [۰]
```

تا

```
testArray [۹]
```

استفاده کنید.

اگر بخواهید به عناصر آرایه خارج از محدوده آن، دسترسی پیدا کنید مثلا،

```
testArray [۱۴]
```

کامپایلر هیچ خطایی نشان نمی دهد و خروجی عدد دلخواه می دهد.

موفق باشید