آرایه (Array) در C++ (جزوه سوم)

گردآورنده : سعید اکبری
استاد راهنما : استاد علیاری

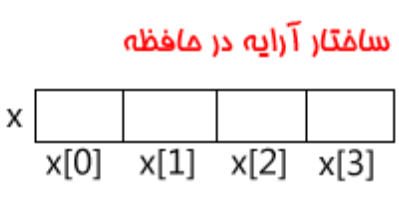
پیش از این ما داده ها را در درون متغیرها و یا ثوابت ذخیره می کردیم. اما گاهی لازم است تعدادی از داده ها را بگونه ای ذخیره نماییم که متعلق به یک مجموعه بوده و دسترسی به آنها آسان باشد.

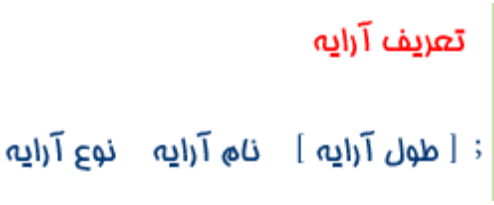
گاهی وقتها لازم است که یک سری از داده هم نوع را دریافت کرده و بروی آنها عملیات خاصی را مثل مقایسه یا مرتب کردن و ... را انجام دهیم، لذا باید آنها را بگونه ای خاص در کنار یکدیگر قرار دهیم که این مسئله در برنامه نویسی را با استفاده از آرایه ها حل خواهیم کرد .

✓ آرایه چیست؟

آرایه مجموعه ای از عناصر هم نوع است. در برنامه نویسی C++ برای تعریف آرایه باید نوع عناصر آنرا مشخص کنیم و آرایه باید حتما دارای نام باشد که از قانون نامگذاری برای متغیرها تبعیت می کند و بعد از نام از [] استفاده می کنیم که درون آن می توان از یک عدد صحیح برای تعیین طول آرایه استفاده نمود .

با هم ساختار یک آرایه را مرور کنیم:





(برای دسترسی به عناصر آرایه از اندیس استفاده می کنیم که اولین اندیس هر آرایه ای از عدد صفر شروع می شود.)

```
type name[Array size]={value1,value2,...};
```

type	نوع آرایه مشخص شده
name	نام آرایه مورد نظر
Array size	اندازه آرایه
value	مقدار های آرایه (عضو های ارایه)

```
int array[4]={2,485,15,58}
```

کد برنامه نویسی C++ بالا آرایه ای را با نام array تعریف می کند که عناصر آن از نوع عدد صحیح هستند و تعداد عناصر آرایه (طول) برابر با ۴ است.

به این نکته توجه کنید که تعداد عناصر آرایه بالا ۴ است ولی اندیس عناصر آن از صفر شروع شده و به سه ختم می شود و نباید تعداد را با اندیس اشتباه گرفت.

گفتنی است که عناصر آرایه پشت سر هم در خانه های حافظه ذخیره می شوند و هر عنصر (خانه) از آرایه به اندازه طول نوع آرایه فضا اشغال می کند. در آرایه بالا چون نوع آرایه تعریف شده int است پس هر عنصر مقدار ۴ بایت و چون طول آن ۴ است در نهایت ۱۶ بایت پشت سر هم از حافظه را اشغال می کند.

معمولا آرایه ها با توجه به ابعادشان تقسیم بندی می شوند:

آرایه های یک بعدی:

آرایه های یک بعدی دارای یک سطر و چند ستون و یا دارای یک ستون و چند سطر هستند و فقط دارای یک اندیس برای دسترسی به عناصرشان می باشند.

مقدار دادن به عناصر آرایه ها:

برای مقدار دادن به آرایه ها هم می توان به تمامی عناصر بصورت یکجا مقدار داد که حالت مجموعه در ریاضی را بخود می گیرد و هم بصورت تک به تک. وقتی طول آرایه را مشخص نمی کنیم با مقدار دادن به عناصر، طول آرایه نیز مشخص خواهد شد.

(نکته) دوستان دقت داشته باشند هنگام تعریف آرایه باید طول را قید کرد مگر اینکه همانجا بدون ذکر طول، به یکایک عناصر مقدار دهیم. پس یا باید طول آرایه را مشخص کنیم یا با مقدار دادن به عنصرهای آرایه،طول برای کامپایلر مشخص شود و اگر غیر از این باشد کامپایلر از برنامه خطا خواهد گرفت .

```
int  x[5] = { 10, 37, -3, 8023, 0 };

////////////// یا \\\\\\\\\\\\

int  x[5];

x[0] = 10;
x[1] = 37;
x[2] = -3;
x[3] = 8023;
x[4] = 0;

////////////// یا \\\\\\\\\\\\

int  x[] = { 10, 37, -3, 8023, 0 };
```

- نکته) همیشه آخرین اندیس آرایه، یک واحد از طول آن کمتر است و تنها دلیل آن شروع شدن اولین اندیس از صفر برای آرایه می باشد.
- برای دسترسی به عناصر آرایه کافی است اندیس آن عنصر از آرایه را در درون [] قید نماییم:

مثال) برنامه ای به زبان C++ بنویسید که تعداد ۵ عدد را از کاربر دریافت کرده و حاصلجمع آنها را در خروجی نمایش دهد:

```
#include <iostream.h>
#include <conio.h>

int main()
{
    int num[5];
    int sum = 0, count;

    for(count = 0 ; count < 5 ; count++)
    {
        cout << "Enter number " << count+1 << " :";
        cin >> num[count];

        sum += num[count];
    }
    cout << "\nSum of numbers is " << sum;
    getch();
    return 0;
}
```

```
Enter number 1: 1
Enter number 2: 5
Enter number 3: 9
Enter number 4: 0
Enter number 5: 18

Sum of numbers is 33
```

در برنامه بالا آرایه ای بنام num با طول ۵ تعریف شده که از یک حلقه تکرار for برای مقدار دادن به عناصر آن استفاده می کنیم و در درون حلقه، مقدار هر عنصر از آرایه را با متغیر sum که دارای مقدار اولیه صفر است جمع می کنیم. دقت کنید که حلقه for باید حتما از صفر شروع شود چون اندیس اولین عنصر آرایه صفر است. در نهایت و با خروج از حلقه حاصلجمع عناصر آرایه که در متغیر sum ریخته شده است را چاپ می کنیم .

ارسال آرایه به عنوان پارامتر برای تابع در برنامه نویسی C++

در این بخش قصد داریم به ارسال آرایه ها به عنوان پارامتر در تابع بپردازیم. در فصل توابع دیدم که متغیرها را به عنوان پارامتر به یک تابع ارسال کردیم، از آرایه ها هم می توان در پارامترهای توابع استفاده نمود. برای اینکار می توان آرایه را با طول ارسال نمود و یا آرایه را بدون طول ارسال کرد و با یک پارامتر دیگری طول را تعیین و ارسال نمود. اما بهتر آن است که آرایه را بدون طول ارسال نماییم و طول را با یک پارامتر دیگر ارسال کنیم، در هر صورت توجه داشته باشید که مانند تعریف آرایه ها هم نام، هم طول و هم نوع آرایه برای تابع و بطور کل برنامه C++ مشخص باشد. به اتفاق هم مثالی را در این زمینه بررسی می کنیم :

```cpp
#include <iostream.h>
#include <conio.h>

int minFunction(int[], int);  //------------------> اعلان تابع با پارامترهای نام آرایه و طول آن

void main()
{
    const int k=4;
    int arr[k];

    cout << "\nMinimum of array elements is " << minFunction(array, k);  //------> فراخوانی تابع
    getch();
}

int mainFunction(int arr[], int length)  //------> تعریف تابع با تمامی متعلقات
{
    for(count = 0 ; count < length ; count++)
    {
        cout << "Enter number [" << count+1 << "] :";
        cin >> arr[cout]
    }

    int minNum = arr[0];

    for(count = 1 ; count < length ; count++)
    {
        if (arr[count] < minNum)
            minNum = arr[cout]
    }

    return minNum;
}
```

```
Enter number 1 : 12
Enter number 2 : 3
Enter number 3 : 28
Enter number 4 : 109

Minimum of array elements is 3
```

همانطور که در کد C++ بالا می بینیم، در اعلان توابع باید نوع و نام آرایه به همراه علامت [] آورده شود تا کامپایلر تشخیص دهد پارامتر ورودی یک تابع است نه عدد صحیح. دومین پارامتر، طول آرایه را به تابع ارسال می کند که اینکار را می توانستیم مستقیما در درون پارامتر اول (آرایه) نیز انجام دهیم.

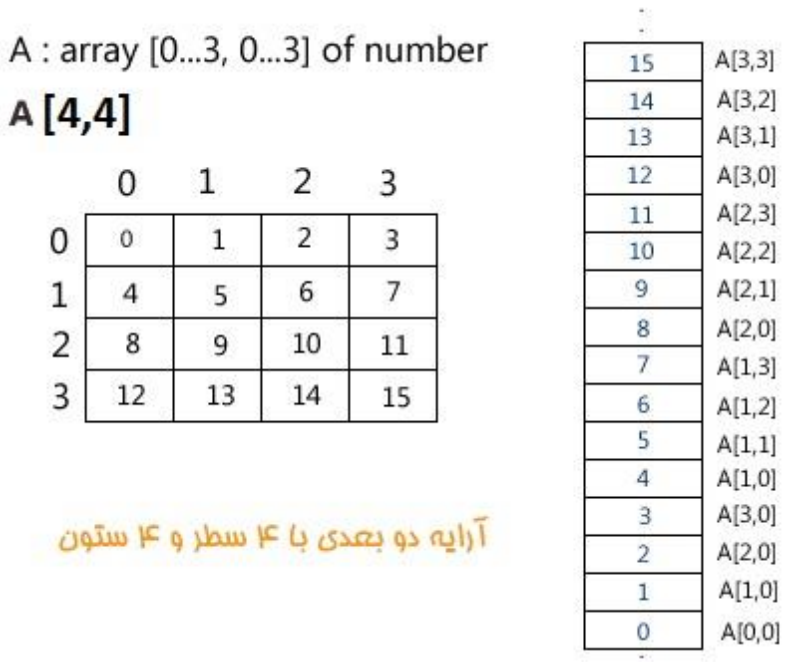
نکته) می بینیم که در فراخوانی تابع که در دستور cout قرار گرفته است ما فقط نام های پارامترها را قید می کنیم و از بیان نوع و [] اجتناب می کنیم.

در نهایت امر، تابع خود را با تمامی پارامترها و انواع و نامهایشان و ملزومات دیگر تعریف می نماییم.

نکته مهم) هر متغیر جز در درون تابع (حوزه) تعریفی خود قابل دسترسی نیست. بعنوان مثال ما در تابع main در کد بالا آرایه array را تعریف کردیم و در و دیگر نمی توانیم از آن در تابع minFunction استفاده کنیم چون کامپایلر محدوده آنرا فقط در تابع خود می داند. اما در تابع minFunction آرایه دیگری را بنام arr تعریف کرده که از آن استفاده می کنیم. این نکته را بخاطر داشته باشید که ما آرایه array را بعنوان پارامتر به تابع ارسال کردیم، و آرایه arr همان آرایه array است اما با نامی دیگر.

برنامه C++بالا با استفاده از توابع و ارسال آرایه بعنوان پارامتر تابع، کوچکترین عدد آرایه را به ما نشان می دهد. برای این کار لازم است ابتدا عناصر آرایه را با یک دستور حلقه تکرار for مقدار دهی کنیم و سپس اولین عنصر را برابر با minNum بگیریم. سپس با یک حلقه for دیگر آنرا با دیگر عناصر آرایه مقایسه می کنیم، اگر مقداری کمتر از آن باشد در متغیر minNum ریخته می شود و اگر نباشد پس خود اولین عنصر از همه کوچکتر است. در دستور حلقه تکرار for دوم به این دلیل count را از ۱ شروع کردیم چون در بالا عنصر با اندیس صفرم آرایه را به عنوان کوچکترین عنصر در نظر گرفتیم و دیگر احتیاجی نیست آنرا دوباره با خودش مقایسه نماییم. عبارت const که در جلوی متغیر k آمده است به کامپایلر می گوید که مقدار این متغیر ثابت است و در طول برنامه تغییری نمی کند و نوشتن هر دستوری مبنی بر تغییر مقدار آن باعث بروز خطا در برنامه خواهد شد.

- آرایه چند بعدی Multi-Dimensional Arrays



آرایه دوبعدی دارای چند سطر و چند ستون می باشد. اولین بعد آرایه بیانگر سطرها و دومین بعد آن بیانگر ستونهاست. همانطور که از شکل بالا پیداست به ترتیب از اولین سطر شروع کرده و تا انتهای ستونهای آن سطر را اندیس گذاری کرده و سپس به سطر بعد رفته و به اندیس گذاری تا آخرین ستون سطر مربوطه می پردازیم و این عمل را تا انتها برای اندیس گذاری تکرار می کنیم. سمت راست شکل هم کاملا گویاست که عناصر آرایه به چه ترتیبی در حافظه کامپیوتر قرار می گیرند. در ادامه به نحوه تعریف این نوع از آرایه و بحث پیرامون آن می پردازیم.

- مقدار دهی به آرایه های چند بعدی:

```
char x[3][4] = {{'t', 'b', 'p', 'z'},   // Row 1
                {'m', 'c', 'c', 'q'},   // Row 2
                {'a', 'z', 'd', 'g'}};  // Row 3

x[0][0] = t;
x[1][3] = q;
x[2][1] = z;
x[2][0] = a;
x[2][3] = g;
```

همانطور که در دستور برنامه نویسی C++بالا مشخص است به ازای هر سطر (بعد اول) یک مجموعه و در درون آن مجموعه، عناصر ستونها (بعد دوم) را مقدار دهی می کنیم. در دستور تعریف آرایه بالا ما ۳ سطر و ۴ ستون داریم که به ازای بعد اول یک مجموعه و به ازای بعد دوم ۴ عنصر در هر مجموعه را تعریف می کنیم.

نکته) عزیزان توجه کنند که چون آرایه ما از نوع char است بنابراین برای هر عنصر باید از ' ' استفاده نماییم، و می دانیم که اگر string باشد باید هر عنصر و بطور کل هر متغیر از این نوع را در " " قرار دهیم. (این یک قانون است).

مثال) برنامه ای با آرایه دو بعدی بنویسید که شماره هر سطر را نوشته و در مقابل آن جمع عناصر آن سطر را هم محاسبه و چاپ نماید:

```
#include <iostream.h>
#include <conio.h>

void sum(int[][2], int);  //------------------> اعلان تابع با پارامترهای نام آرایه و طول آن

void main()
{
    const int m=3, n=2;
    int matrix[m][n] = {{2,5}, {15,9}, {0,32}};

    sum(matrix, m);
    getch();
}

void sum(int x[][2], int a)
{
    int i, j;
    cout << "Row\t\t" << "Sum\t";
    cout << "---------------------\n";

    for (i=0 ; i<a ; i++)
    {
        int sum = 0;
        for (j=0 ; j<2 ; j++)
            sum += x[i][j];
        cout << i+i << "\t\t" << sum << "\n";
    }
}
```

```
Row        Sum
---------------
1           7
2           24
3           32
```