

برنامه‌سازی کامپیووتر

جلسه هشتم

آرایه‌ها در زبان C

طرح کلی

- آرایه‌ها
 - آرایه‌های یک بعدی
 - آرایه‌های چندبعدی
 - آرایه‌ها به عنوان پارامتر
 - مقداردهی اولیه

مقدمه

- در زبان C با استفاده از متغیرها و ثوابت می‌توانیم داده‌ها را ذخیره کرده و در صورت لزوم از آن‌ها استفاده کنیم. در برخی موارد نیازمند ذخیره تعداد زیادی داده هستیم.
- نمرات و معدل دانشجویان یک کلاس، حقوق کارمندان یک شرکت، موجودی کالاهای یک انبار و بسیاری موارد دیگر از جمله مواردی هستند که نیازمند استفاده از تعداد زیادی متغیر برای ذخیره و بازیابی اطلاعات هستند.
- در این موارد تعریف و بکارگیری متغیرها به روش معمول کاری بسیار طاقت‌فرسا و پیچیده است. لذا از مجموعه‌ای از متغیرها با نام **آرایه** استفاده می‌کنیم.
- **آرایه مجموعه‌ای از عناصر همنوع است.**
- هر آرایه دارای نامی است که مانند متغیرهای معمولی نام‌گذاری می‌شود.
- برای دسترسی به عناصر آرایه از متغیری به نام **اندیس** استفاده می‌شود. به همین دلیل آرایه را متغیر **اندیس‌دار** نیز می‌گویند.

آرایه‌های یک بعدی (۱)

آرایه‌های یک بعدی را لیست نیز می‌نامند و بصورت زیر تعریف می‌شوند.

[طول آرایه] نام آرایه نوع آرایه

- نوع آرایه یکی از انواع قابل قبول در زبان C است.
- نام آرایه، برای دسترسی به عناصر آرایه مورد استفاده قرار می‌گیرد.
- طول آرایه با یک عدد صحیح مثبت و در همان ابتدای تعریف آرایه مشخص می‌شود.

مثال:

int x[5];

آرایه‌ای با نام x و به طول ۵ از نوع int تعریف می‌کند.

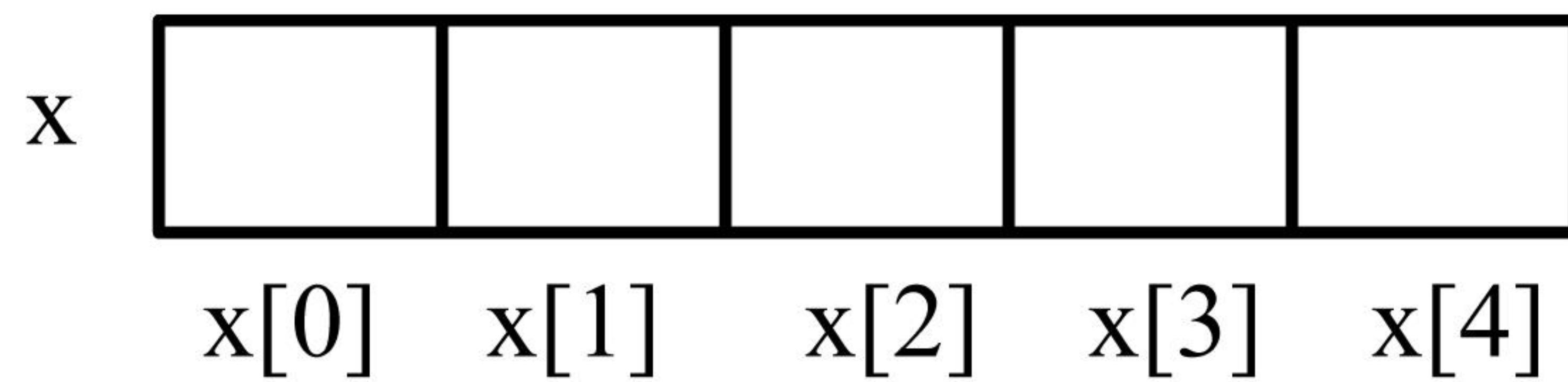
float y[100];
double z[10];
char c[32];

آرایه‌های یک بعدی (۲)

- اندیس آرایه‌ها در زبان C از صفر شروع می‌شود.
- اندیس آرایه‌ای به طول n از صفر شروع شده و به $n-1$ ختم می‌شود.
- عناصر آرایه در محل‌های متوالی از حافظه ذخیره می‌شوند.

: مثال :

```
int x[5];
```



- میزان حافظه‌ای که به آرایه اختصاص داده می‌شود از رابطه زیر قابل محاسبه است.
$$(\text{طول آرایه}) * (\text{طول نوع آرایه}) = \text{میزان حافظه آرایه}$$
- برای مثال اگر طول هر int را ۴ بایت درنظر بگیریم، میزان حافظه‌ای که آرایه x اشغال می‌کند، برابر با $20 = 5 * 4$ بایت خواهد بود.

آرایه‌های یک بعدی (۳)

- با استفاده از اندیس آرایه می‌توان به عناصر آرایه دسترسی پیدا کرد.
مثال:

▪ دستور زیر مقدار موجود در خانه‌ی سوم آرایه x را چاپ می‌کند.

```
printf ("%d", x[2]);
```

▪ دستور انتساب زیر مقدار ۲۳ را در عنصر اول آرایه x قرار می‌دهد.

```
x[0] = 23;
```

```
int x[5], i;  
for (i=0;i<5;i++)  
    scanf("%d", &x[i]);  
  
for (i=0;i<5;i++)  
    printf("%d\n", x[i]);  
  
for (i=4;i>=0;i--)  
    printf("%d\n", x[i]);
```

- برای استفاده راحت‌تر از آرایه‌ها معمولاً از حلقه‌های تکرار استفاده می‌کنند.

مثال ۱

مثال : برنامه‌ای بنویسید که ۵ عدد را گرفته و در آرایه‌ای قرار می‌دهد و مجموع آن‌ها را محاسبه کرده و چاپ می‌کند.

```
#include <stdio.h>
int main()
{
    int x[5], i, sum=0;
    printf("Enter 5 numbers:\n");
    for (i=0;i<=4;i++)
        scanf("%d",&x[i]);
    for (i=0;i<=4;i++)
        sum=sum+x[i];
    printf("Sum of your numbers is: %d", sum);
    getch();
    return 0;
}
```

```
Enter 5 numbers:
12 32 7 22 1
Sum of your numbers is: 74
```

مثال ۲

مثال : برنامه‌ای که ۵ عدد را گرفته و به ترتیب در آرایه‌ای ذخیره می‌کند. آرایه را بصورت معکوس چاپ می‌کند.

```
#include <stdio.h>
int main()
{
    int x[5], i;

    printf("Enter 5 numbers:\n");
    for (i=0;i<=4;i++)
        scanf("%d",&x[i]);

    printf("Inverse of your numbers:\n");
    for (i=4;i>=0;i--)
        printf("%d ", x[i]);

    getch();
    return 0;
}
```

```
Enter 5 numbers:
12 0 93 45 66
Inverse of your numbers:
66 45 93 0 12
```

مثال ۳

مثال : برنامه‌ای بنویسید که معدل ۵ دانشجو را دریافت کرده و در آرایه‌ای قرار می‌دهد و بیشترین معدل و محل وجود آن را پیدا می‌کند و در خروجی نمایش می‌دهد.

```
#include <stdio.h>

int main()
{
    const int n=5;
    float ave[n], amax =0;
    int i, p;

    for (i=0;i<n;i++)
    {
        printf("enter an average:");
        scanf("%f",&ave[i]);
    }

    amax = ave[0];
    p=0;
```

```
for (i=1;i<n;i++)
{
    if (ave[i] > amax)
    {
        amax=ave[i];
        p=i;
    }
}

printf("\n max = %5.2f, position = %d", amax, p+1);
getch();
return 0;
```

```
enter an average:11
enter an average:17
enter an average:19
enter an average:12
enter an average:10

max = 19.00, position = 3
```

ایه یک بعدی به عنوان آرگومان تابع

دانشگ

```
void func1 (int x[]);
void func2 (int x[], int len);
int main()
{
    int x[10];
    ...
    func1(x);
    ...
    func2(x, 10);
    return 0;
}
```

```
void func1 (int x[10])
{
```

```
    ....
```

```
}
```

```
void func2 (int x[], int len)
{
```

```
    ....
```

```
}
```

- برای ارسال آرایه به تابع، باید **نام آرایه** به عنوان آرگومان ذکر شود.

- اگر آرایه به عنوان آرگومان تابع باشد، **پارامتر معادل آن** می‌تواند به سه صورت زیر تعریف شود:

- آرایه‌ای با طول مشخص

- آرایه‌ای با طول نامشخص که در این صورت بهتر است طول آرایه به عنوان آرگومانی دیگر به تابع ارسال شود.

- اشاره گر

آرایه‌های دو بعدی (۱)

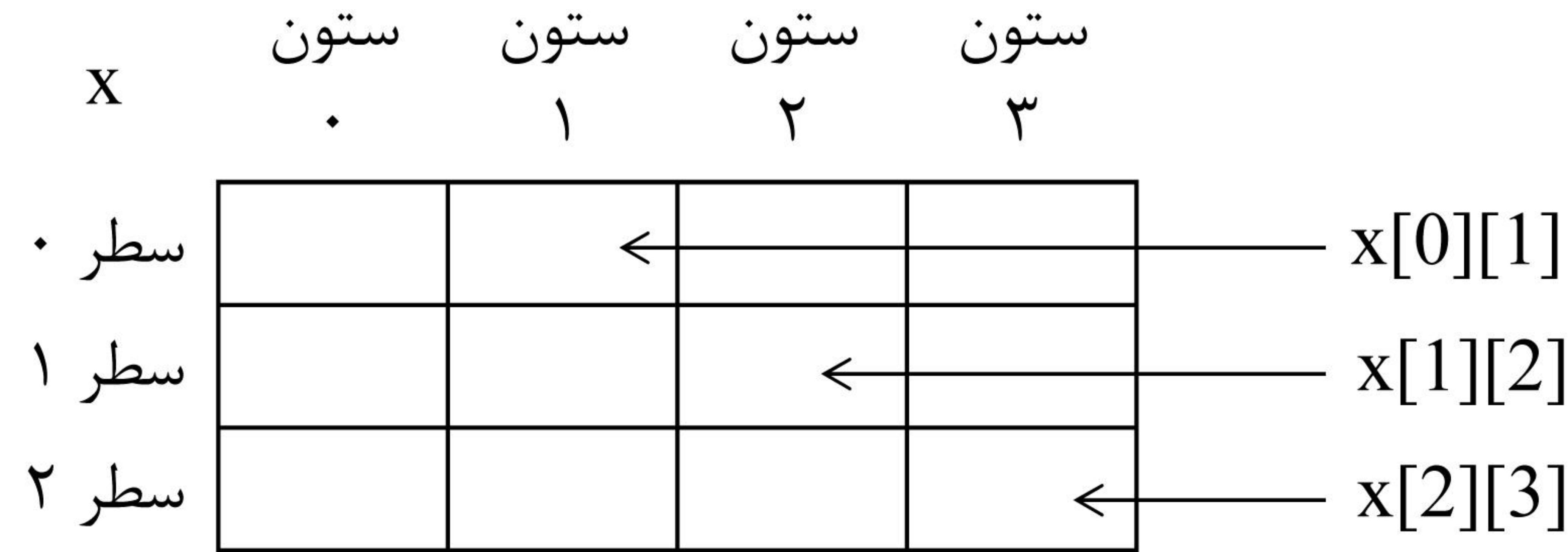
- علاوه بر لیست‌ها (آرایه‌های یک بعدی)، می‌توان آرایه‌هایی تعریف کرد که بیش از یک بعد داشته باشند.
 - مثال:
- جدول ضرب آرایه‌ای دو بعدی است که یک بعد آن سطر و بعد دیگر آن ستون است.
- ماتریس‌ها
- در آرایه‌های دو بعدی، برای دستیابی به عناصر هر آرایه، از دو اندیس استفاده می‌شود که اندیس اول را **سطر** و اندیس دوم را **ستون** می‌نامند.
- آرایه‌هایی با بیش از دو بعد (آرایه‌های n بعدی) قابل استفاده‌اند.
- آرایه دو بعدی در C به صورت زیر تعریف می‌شود:

[**بعد ۱**] [**بعد ۲**];
- هر کدام از این اندیس‌ها از صفر شروع می‌شوند.

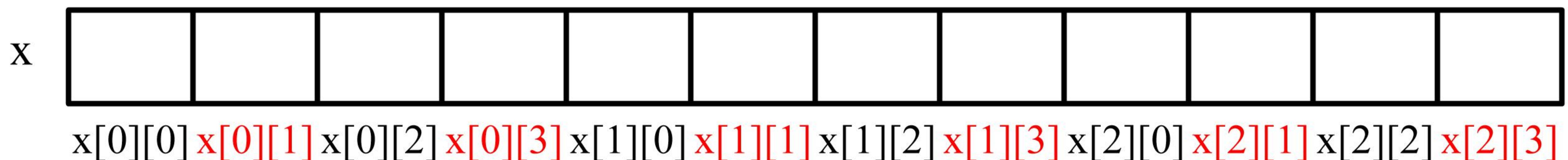
آرایه‌های دو بعدی (۲)

مثال:

```
int x[3][4];
```



- آرایه‌ها در حافظه به صورت سطري و ستوني می‌توانند ذخیره شوند.
- در زبان C آرایه‌ها به صورت سطري ذخیره می‌شوند. یعنی ابتدا عناصر سطر اول، سپس عناصر سطر دوم، عناصر سطر سوم و ... ذخیره می‌شوند.
- برای مثال قبل می‌توان آرایه را در حافظه به این صورت نمایش داد.



مثال

دانشگ

- مثال : برنامه‌ای بنویسید که جدول ضرب اعداد را با استفاده از آرایه‌های دو بعدی تولید کرده و نمایش دهد.

```
#include <stdio.h>

int main()
{
    int table[10][10], i , j;
    for (i=0;i<10;i++)
        for (j=0;j<10;j++)
            table[i][j]=(i+1)*(j+1);

    for (i=0;i<10;i++)
    {for (j=0;j<10;j++)
        printf("%4d", table[i][j]);
        printf("\n");}
    getch();
    return 0;
}
```

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|-----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
| 3 | 6 | 9 | 12 | 15 | 18 | 21 | 24 | 27 | 30 |
| 4 | 8 | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 |
| 5 | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45 | 50 |
| 6 | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54 | 60 |
| 7 | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63 | 70 |
| 8 | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72 | 80 |
| 9 | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81 | 90 |
| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |

ایه‌های دو بعدی به عنوان آرگومان توابع

دانشگ

```
void func1 (int x[5][10]);
void func2 (int x[][10], int len);
int main()
{
    int x[5][10];
    ...
    func1(x);
    ...
    func2(x, 5);
    return 0;
}
```

```
void func1 (int x[5][10])
```

```
{  
    ...  
}
```

```
void func2 (int x[][10], int len)
```

```
{  
    ...  
}
```

- برای ارسال آرایه دو بعدی از تابعی به تابع دیگر، باید نام آرایه را به عنوان آرگومان تابع ذکر کرد.

برای تعریف پارامتر معادل آن، مانند آرایه‌های یک بعدی عمل می‌شود. با این تفاوت که در حالتی که در آرایه‌های یک بعدی، پارامتر به صورت آرایه بدون طول ذکر می‌شد، در آرایه دو بعدی، طول سطر ذکر نمی‌شود ولی طول ستون حتماً باید ذکر شود که در این صورت بهتر است، طول سطر، به عنوان آرگومان دیگری به تابع ارسال شود.

- به هنگام ارسال آرایه به عنوان پارامتر در آرایه‌های چندبعدی نیز تنها می‌توان اندیس اول را حذف کرد و عدد مربوط به بقیه اندیس‌ها باید صراحتاً ذکر شود.

مثال

```
#include <stdio.h>
void printTable(int table[][10]);
int main()
{
    int table[10][10], i , j;
    for (i=0;i<10;i++)
        for (j=0;j<10;j++)
            table[i][j]=(i+1)*(j+1);
    printTable(table);
    getch();
    return 0;
}
```

```
void printTable(int table[][10])
{
    int i,j;
    for (i=0;i<10;i++)
    {
        for (j=0;j<10;j++)
            printf("%4d", table[i][j]);
        printf("\n");
    }
}
```

مقدار اولیه آرایه‌ها (۱)

- اگر آرایه‌ای خارج از توابع تعریف شود (آرایه عمومی)، مقدار اولیه عناصر آن صفر خواهد بود.
- اگر آرایه در داخل تابعی تعریف شود، مقدار اولیه عناصر آن نامعلوم خواهد بود.
- بصورت زیر می‌توان به آرایه‌ها در هنگام تعریف آن‌ها مقدار اولیه داد.

نام آرایه نوع آرایه ; {مقادیر} = [بعد ۱] [بعد ۲] [...] [بعد ۵]

مثال:

• آرایه صحیح ۴ عنصری با مقادیر ۵، ۰، ۳، ۲ را تعریف می‌کند.

```
int x[4] = {5, 0, 3, 2};
```

| | | | | |
|---|---|---|---|---|
| x | 5 | 0 | 3 | 2 |
|---|---|---|---|---|

• مقداردهی به آرایه دو بعدی به دو روش:

```
int y[2][3] = {1, 3, 4, 7, 6, 15}
```

```
int y[2][3] = {{1, 3, 4}, {7, 6, 15}}
```

مقدار اولیه آرایه‌ها (۲)

- اگر هنگام تعریف آرایه، مقدار اولیه به عناصر آرایه داده شود، اما طول آرایه مشخص نشود، تعداد عناصر آرایه به اندازه تعداد مقادیری است که به آن نسبت داده می‌شود.
مثال : آرایه‌ای به طول ۳ تعریف می‌شود.

```
int x [] = { 1, 5, 3}
```

- اگر تعداد مقادیری که به آرایه نسبت داده می‌شود کمتر از طول آرایه باشد، عناصری که مقدار اولیه‌ای برای آن‌ها تعیین نشده است با مقدار صفر مقداردهی می‌شوند.
مثال : عنصر اول با مقدار ۱۲، عنصر دوم با مقدار ۴۳، عنصر سوم با مقدار ۳ و بقیه عناصر با مقدار صفر مقداردهی می‌شوند.

```
int x[10] = {12, 43, 3};
```

مثال آرایه چندبعدی:

```
int x [3][2][4] = { {{5,6,8,5},{2,3,5,8}},  
                     {{7,9,0,3},{8,1,3,0}},  
                     {{3,7,5,9},{9,9,9,2}} };
```

نکته

- در زبان C، هنگام تعریف آرایه باید طول آن مشخص باشد.
- در برخی مسائل ممکن است تعداد دقیق عناصری که باید در آرایه قرار گیرند مشخص نشده باشد. در این صورت باید حداکثر تعدادی را که ممکن است مورد نیاز باشد را در نظر بگیریم.
- برای راحتی کار بهتر است طول آرایه را بصورت یک ثابت در برنامه تعریف کنیم و در تمامی مواردی که موردنیاز است از آن ثابت استفاده کنیم.
- هنگام ارسال پارامترها طول آرایه‌ها یا در تعریف تابع ذکر شود و یا بصورت یک پارامتر جداگانه ارسال گردد.
- همیشه توجه داشته باشیم که اندیس آرایه‌ها از صفر شروع می‌شود.
- به هیچ وجه اندیس آرایه‌ها نباید از محدوده‌ی مجازی که برای آرایه تعریف شده است تجاوز کند.

تمرینات تکمیلی

- برنامه‌های زیر را به ترتیب با استفاده از یکی از کامپایلرهای زبان C طراحی، پیاده‌سازی و اجرا کنید.
 1. برنامه‌ای بنویسید که ۱۰ عدد صحیح را از ورودی گرفته و در آرایه‌ای قرار داده و سپس آن‌ها را به ترتیب نمایش دهد.
 2. برنامه‌ای بنویسید که ۱۰ عدد را گرفته و مجموع و میانگین آن‌ها را محاسبه کرده و نمایش دهد.
 3. برنامه‌ای بنویسید که دو بردار را گرفته و مجموع آن دو بردار را نمایش دهد.
 4. برنامه‌ای بنویسید که دو ماتریس 10×10 را گرفته و مجموع آن دو ماتریس را نمایش دهد.
 5. تابعی بنویسید که یک آرایه و طول آن را گرفته و مجموع عناصر آن آرایه را برگرداند.
 6. برنامه‌ای بنویسید که ۱۰۰ عدد از اعداد فیبوناچی را به ترتیب در آرایه‌ای ۱۰۰ تایی قرار دهد.

جمع‌بندی

- آرایه‌ها

- آرایه‌های یک بعدی

- آرایه‌های چندبعدی

- آرایه‌ها به عنوان پارامتر

- مقداردهی اولیه

- رشته‌ها

تمرینات فصل پنجم کتاب

۱۱-۶-۳-۲-۱