

## جلسه پنجم (آرایه ها)

آرایه ها تعدادی متغیر هم نوع هستند که با یک نام مشترک و یک عدد بنام اندکس منحصر به فرد برای هر متغیر شناخته می شوند. حافظه مربوط به این متغیرها بصورت متوالی و پشت سرهم در حافظه قرار میگیرند. وبه صورت

تعریف می شوند.

مثال:

```
int x[10];
float y[4];
char z[30];
```

### نکته:

1. اندکس اولین خانه از صفر شروع می شود.

2. در صورت دستیابی به خانه های تعریف نشده از یک آرایه مثلا  $x[10]$  پیغام خطایی دریافت نخواهید کرد. لذا همیشه مراقب بازه اندکس ها باشید چرا که چنین دستیابی های غیر مجازی می تواند نتایج عجیب و غیر قابل پیش بینی داشته باشد مثل تغییر مقادیر سایر متغیرها.

بعد از تعریف یک آرایه شما می توانید با عباراتی مثل  $x[0]$   $x[1]$  ...  $x[n-1]$  به عناصر یا همان متغیرهای آرایه دسترسی داشته باشید در واقع عبارات فوق کاملا مشابه متغیرهای معمولی دیگر عمل میکنند.

برای مثال می توانید بنویسید:

```
cin>>x[0];
```

با این دستور می توانید اولین عنصر آرایه را بخوانید. یا با قرار دادن این دستور در یک حلقه for و استفاده از اندکس حلقه بجای عدد صفر تمام عناصر آرایه را بخوانید.

```
for(int i=0;i<10;i++)
    cin>>x[i];
```

### نکته:

1. انتساب آرایه ها به یکدیگر غیر مجاز است و سبب بروز خطای کامپایلری می شود.

```
int x[10];
int y[10];
x=y;
```

خطا

2. از آرایه های کاراکتری برای ذخیره ی رشته ها استفاده می شود که در جلسات بعدی بیان خواهد شد.

3. آرایه ها می توانند چند بعدی باشند مثل یک ماتریس دو بعدی یا یک مکعب سه بعدی که تجسمش کمی مشکل است. معمولا برنامه نویسان از آرایه های بیشتر از دوبعدی استفاده نمی کنند.

```
int a[3][4][5];
```

### مقدار اولیه در آرایه ها:

آرایه ها مانند سایر انواع مقدار اولیه ندارند لذا باید در صورت لزوم به آنها مقدار اولیه داد.

به مثال ها توجه کنید.

```
int x[5]={2,56,4,8,4};
```

پنج خانه آرایه بترتیب با اعداد ذکر شده مقدار دهی میشوند.

```
int y[2][3]={1,4,5,9,54,21,97};
```

```
int y[2][3]={{1,4,5},{54,21,97}};
```

دو دستور بالا یک کار برد دارند ام خوانایی دستور دوم بیشتر است.

```
int y3d[2][3][4]={{1,2,3,4},{5,6,7,8},{9,10,11,12}},  
                {{13,14,15,16},{17,18,19,20},{21,22,23,24}}};
```

برای آرایه های چند بعدی بهتر است برای خوانایی بیشتر مقادیر هر سطر را در یک آکولاد قرار دهید.

```
int z[]={1,2,3};
```

در صورتی که اندازه آرایه ذکر نشود برابر تعداد مقادیر ذکر شده در نظر گرفته می شود.

```
int x[5]={1,2};
```

```
int y[2][3]={1,4,5,9};
```

```
int y3d[2][3][4]={};
```

در صورتی که تعداد مقادیر کمتر از تعداد خانه ها باشد سایر خانه ها مقدار صفر میگیرند.

### نکته:

آرایه ی عمومی همیشه مقدار اولیه صفر دارد. یعنی تمام خانه های آن صفر اند.

cpplearning.blogspot.com