



مبانی کامپیوتر و برنامه سازی

فصل هشتم: آرایه ها

فهرست مطالب

- تعریف آرایه
- مقدار دهی آرایه
- ارسال آرایه به تابع
- الگوریتم جستجوی خطی
- مرتب‌سازی انتخابی
- الگوریتم جستجوی دودویی
- آرایه‌های چند بعدی



انگیزه: آیا می‌توانیم واریانس ۱۰ عدد را محاسبه کنیم!؟

- آیا می‌توانید برنامه‌ای بنویسید که 10 عدد را گرفته و میانگین و واریانس آنها را محاسبه کند؟ محاسبه میانگین کار ساده‌ای می‌باشد؛ می‌توان نوشت

```
float X, Sum = 0, Avg;  
for (i = 0; i < 10; i++)  
    { scanf("%f", &X); Sum += X; }  
Avg = Sum / 10;
```

- اما برای محاسبه واریانس باید

1. میانگین اعداد را بدست آورد و

2. مربع تفاضل هر یک تا میانگین را محاسبه نمود!

- می‌توان دید که دو بار باید به اعداد مراجعه نمود، یعنی باید آنها را به خاطر سپرد؛ برای انجام این کار باید تمامی اعداد را در حافظه نگه‌داری نمود. در قطعه برنامه نوشته شده ما فقط آخرین عدد، یعنی X ، را در حافظه داریم. (با تعریف متغیرهای X_1, X_2, \dots, X_{10} می‌توان مشکل را حل نمود، اما برای محاسبه واریانس 1000 عدد نمی‌توان از این روش استفاده نمود 😊).

با استفاده از آنچه تا به حال یاد گرفته‌ایم نمی‌توانیم بسیاری از محاسبات عمومی مانند: محاسبه واریانس، مرتب سازی، پیدا کردن یک عدد، و ... را انجام دهیم.

وجه مشترک تمام اعمال فوق نیاز به **ذخیره سازی تعداد زیاد (و یا نامعلوم) داده یکسان در حافظه است.**

تعریف آرایه

آرایه، یک تعداد متغیرها هم نوع است که پشت سر هم در حافظه قرار گرفته‌اند. به هر یک از این متغیرها، یک مولفه آرایه گفته می‌شود.

- برای نمونه یک آرایه با ده عنصر از نوع `int`.
- برای دسترسی به هر عنصر آرایه باید شماره آنرا بدانیم. در ریاضیات برای مشخص کردن شماره (یا اندیس) یک عنصر، آن شماره را در پایین نام آرایه می‌نویسیم: A_3 یعنی سومین عنصر در آرایه با نام `A`.

تعریف آرایه در زبان C

• تعریف یک آرایه در زبان C:

• `type name [size];`

• مثال:

• `int X[10];` (یک آرایه 10 عنصری با نام X)

• `float Y[20], Z[20];`

• برای دسترسی به عناصر آرایه باید اندیس مورد نظر را داخل [] قرار دهیم. مثلاً `X[2]` سومین عنصر آرایه را در دسترسی قرار می‌دهد. **عناصر آرایه در زبان C از شماره صفر شروع می‌شوند.** نمونه‌ها:

• `X[2]=7;`

• `printf("%d", X[5]);`

• در حقیقت آرایه چیزی جز چندین خانه پشت سر هم در حافظه نیست. در مثال اول ده خانه از نوع صحیح به ما اختصاص داده شده است و می‌توانیم به هر کدام از این خانه‌ها مقداری داده یا مقدار آنها را بدست بیاوریم.

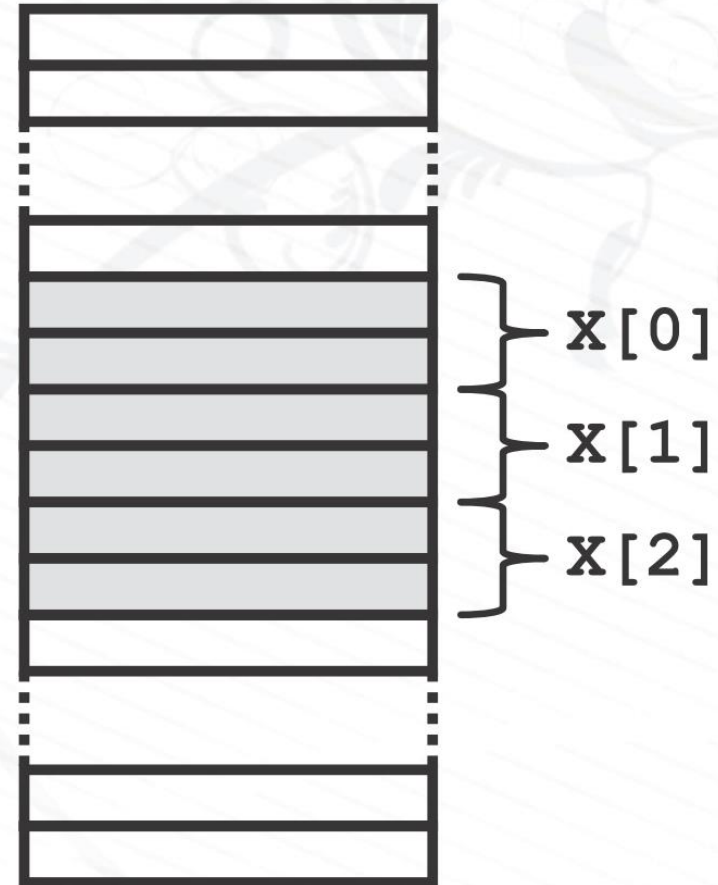
اندیس آرایه

- عنصر اول هر آرایه عنصر 0 ام است! یا ایندیس عنصر اول 0 است.
- بنابراین، آخرین عنصر یک آرایه n عنصری در محل $n-1$ ام قرار دارد.
- سعی در خواندن یا نوشتن عنصری با اندیس بالاتر از $n-1$ نتیجه

نحوه قرار گرفتن عناصر آرایه در حافظه

RAM

```
int X[3];
```



می‌دانیم! که هر متغیر از نوع `int` دو بایت حافظه را اشغال می‌کند.


```
#include <stdio.h>
main()
{
    double A[10];
    A[0]=1;
    A[1]=2;
    A[2]=4;
    A[3]=8;
    printf("%f", A[0]+A[1]+A[2]+A[3]);
    for (i = 0; i < 10; i++)
        A[i] = i*2;
}
```

```
#include <stdio.h>
main()
{
    double A[10];
    int i;

    for (i = 0; i < 5; i++)
        scanf("%f", &A[i]);

    for (i = 0; i < 10; i++)
        printf("%f ", A[i]);
}
```

استفاده از متغیری صحیح جهت در دسترس قرار دادن خانه‌ها برای خواندن یا نوشتن امری متداول است. به چنین متغیرهای **شمارنده** می‌گویند.

این حلقه پنج عنصر اول آرایه را از کاربر می‌گیرد.

این حلقه ده عنصر اول آرایه را نمایش می‌دهد. مقدار پنج عنصر انتهایی نامشخص است!

دادن مقدار به آرایه در هنگام تعریف

• برای دادن مقدار به آرایه در هنگام تعریف باید مقادیر عناصر را داخل {} نوشت. نمونه‌ها:

• `double A[10]={1, 2};`

به دو عنصر اول به ترتیب 1 و 2 و داده شده و مابقی 0 خواهند بود.
`int x[100]={0}`

• `int b[100]={0};`

یک آرایه 100 عنصری با نام b تعریف می‌کند که تمام 100 عنصر آن 0 هستند.

• `int b[]={3,4,5};`

یک آرایه سه عنصری که مقادیر هر سه درایه آن داده شده است. اگر اندازه آرایه هنگام تعریف داده نشود زبان C آن را برابر تعداد مقادیر داده شده در نظر می‌گیرد.

Int i=8; ●

Int x[i]; ●

اندازه آرایه باید یک عدد ثابت باشد. بنابراین نوشتن به صورت بالا نادرست است.

قطعه برنامه‌ای برای محاسبه میانگین یک آرایه داده شده

```
int Sum;  
for (i = 0; i < n; i++)  
    Sum += X[i];  
Sum /= n;
```

قطعه برنامه‌ای برای محاسبه میانگین یک آرایه داده شده

```
#include <stdio.h>
void main()
{
    float X[1000], Y[1000], Z[1000];
    int i, n;
    scanf("%d", &n);

    for (i = 0; i < n; i++)
    {
        printf("X(%d):", i);
        scanf("%f", &X[i]);
    }

    for (i = 0; i < n; i++)
    {
        printf("Y(%d):", i);
        scanf("%f", &Y[i]);
    }

    for (i = 0; i < n; i++)
        Z[i] = X[i] + Y[i];

    for (i = 0; i < n; i++)
        printf("Z(%d)=%f ", i, Z[i]);
}
```