ساختمان داده ها در زبان C

Data Structure in C

میلاد سلطانی

فصل دوم

آرایه ها

مقدمه

- ساده ترین ساختمان داده در زبان C آرایه است.
- آرایه در ساده ترین شکل خود به صورت یک آرایه یک بعدی است که بردار نامیده می شود.
- آرایه یک بعدی مجموعه ای مرتب و محدودی از عناصر همگن است.
  - ✯ مرتب : عناصر آرایه دارای ترتیب خاصی هستند.
  - ✯ محدود : تعداد عناصر آرایه مشخص است، ممکن است بزرگ یا کوچک باشد.
  - ✯ همگن : کلیه عناصر آرایه از یک نوع هستند.

آرایه به عنوان یک ADT

- آرایه را می توان بصورت یک نوع داده انتزاعی در نظر گرفت :

ADT آرایه

مجموعه عناصر :

دنباله با طول ثابت (مجموعه مرتب) از عناصر همگی از یک نوع هستند.

عملیات اصلی :

دستیابی مستقیم به هر عنصر آرایه، به طوری که مقادیر را می توان از این عنصر بازیابی یا در آن ذخیره کرد.

آرایه به عنوان یک ADT

- معنای دستیابی مستقیم یا تصادفی این است که هر عنصر آرایه می تواند با مشخص کردن محل آن در آرایه دستیابی شود، به طوری که زمان دستیابی به هر عنصر آرایه یکسان است و به موقعیت آن در آرایه بستگی ندارد.

- معنای دستیابی ترتیبی این است که دستیابی به یک عنصر زمانی امکان پذیر است که عناصر قبلی دستیابی شوند.

آرایه های یک بعدی

- آرایه یک بعدی به صورت زیر تعریف می شود :

اعلان آرایه در C

شکل کلی :

```
نوع name[CAPACITY];
نوع name[CAPACITY] = {مقادیر اولیه};
```

نوع یک از انواع متداول در زبان برنامه نویسی C است و name نام آرایه است که مثل یک متغیر معمولی تعیین می شود. CAPACITY تعداد مقادیری است که می تواند در آرایه ذخیره شود.

در حالت دوم (خط دوم) تعریف آرایه که به عناصر آرایه مقدار اولیه داده می شود، می توان از CAPACITY صرف نظر کرد.

هدف :

به کامپایلر گفته می شود که بلوکی از محل های متوالی حافظه را برای نگهداری تعداد CAPACITY عنصر از نوع مشخص شده در نظر بگیرد و نام آن محل را name بگذارد.

آرایه های یک بعدی

- با توجه به توضیحات داده شده، می توان دریافت که ویژگی های آرایه در زبان C تعریف یک آرایه را به صورت ADT پیاده سازی می کند :

آرایه به عنوان ADT	آرایه در زبان C
اندازه ثابت	CAPACITY ظرفیت آرایه را مشخص می کند
ترتیب	اندیس ها از 0 تا CAPACITY – 1 شماره گذاری می شوند
عناصر همنوع	نوع مشخص کننده نوع عناصر آرایه است
دستیابی مستقیم	عمگر اندیس[]

- نمونه ای از تعریف آرایه در زبان C :

```
1) int  a[10];
2) float  p[5];
3) int  [ ] = {0,11,21,63};
4) char  ch[13];
```

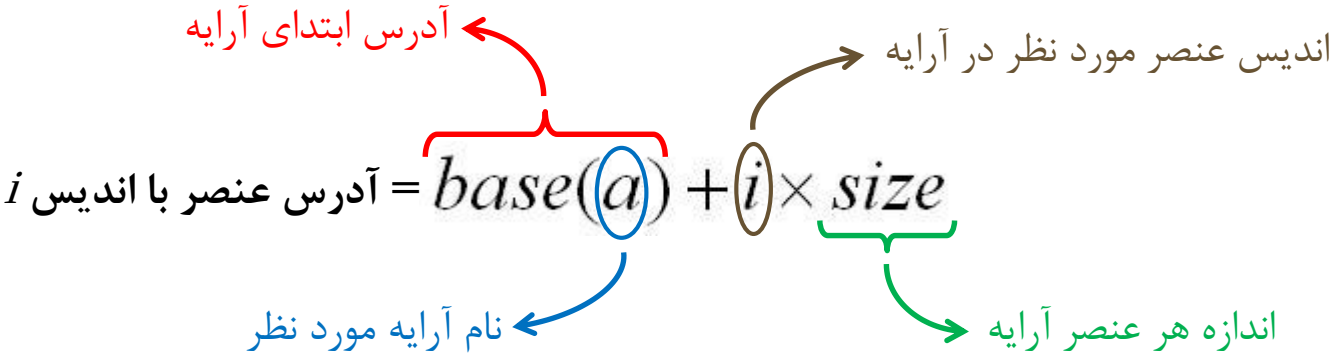
پیاده سازی آرایه یک بعدی

- دستور زیر به آسانی یک آرایه یک بعدی را پیاده سازی می کند :

```
int  a[10];
```

✓ این دستور ۱۰ مکان متوالی حافظه را که هر کدام محل ذخیره سازی یک عدد صحیح است، برای یک برنامه تخصیص می دهد.

- اگر بخواهیم آدرس فیزیکی محل ذخیره سازی داده مورد نظر را بیابیم، از رابطه زیر استفاده می کنیم:



نکته! اندیس هر عنصر در آرایه یک واحد کمتر از شماره عنصر در آرایه است.

آرایه های دو بعدی

- آرایه دو بعدی به صورت زیر تعریف می شود :

اعلان آرایه دو بعدی در C

شکل کلی:

نوع name[DIM_1][DIM_2];

نوع name[DIM1][DIM2] = {مقادیر اولیه};

DIM_1 و DIM_2 به ترتیب اندیس های سطر و ستون آرایه هستند.

در حالت دوم عناصر آرایه بیان کننده تعداد عناصر و/ یا مقدار عناصر آرایه هستند.

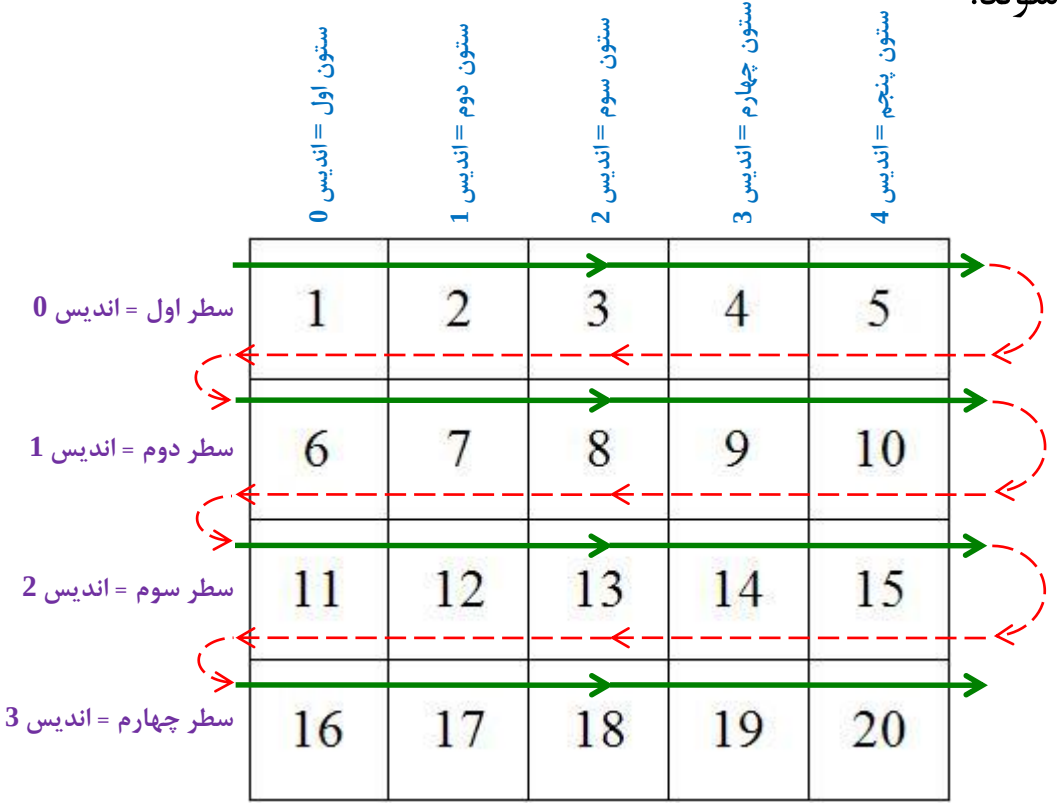
هدف:

به تعداد $DIM_1 \times DIM_2$ محل متوالی از حافظه سیستم به عناصر آرایه تخصیص می یابد.

پیاده سازی آرایه دو بعدی

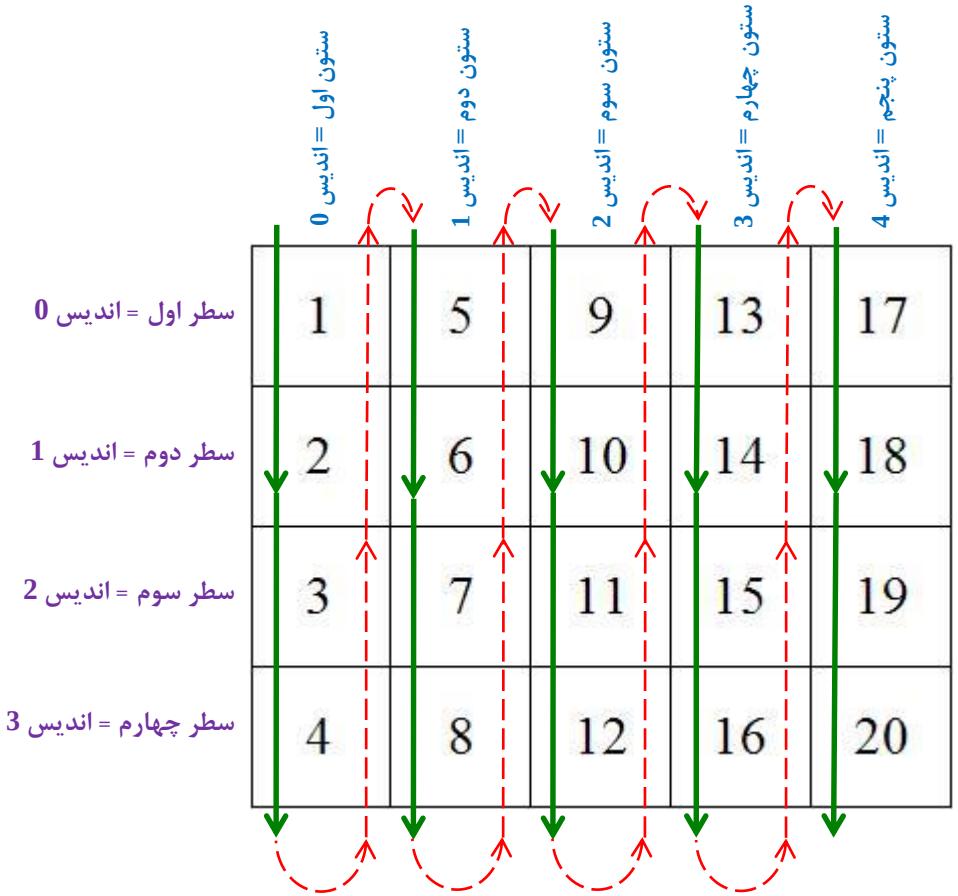
- آرایه های دو بعدی در حافظه به دو صورت ذخیره می شوند.

A. ذخیره سازی سطری : در این حالت عناصر آرایه سطر به سطر پشت سر هم بصورت متوالی در حافظه ذخیره می شوند.



پیاده سازی آرایه دو بعدی

- B. ذخیره سازی ستونی : در این حالت عناصر آرایه ستون به ستون پشت سر هم بصورت متوالی در حافظه ذخیره می شوند.



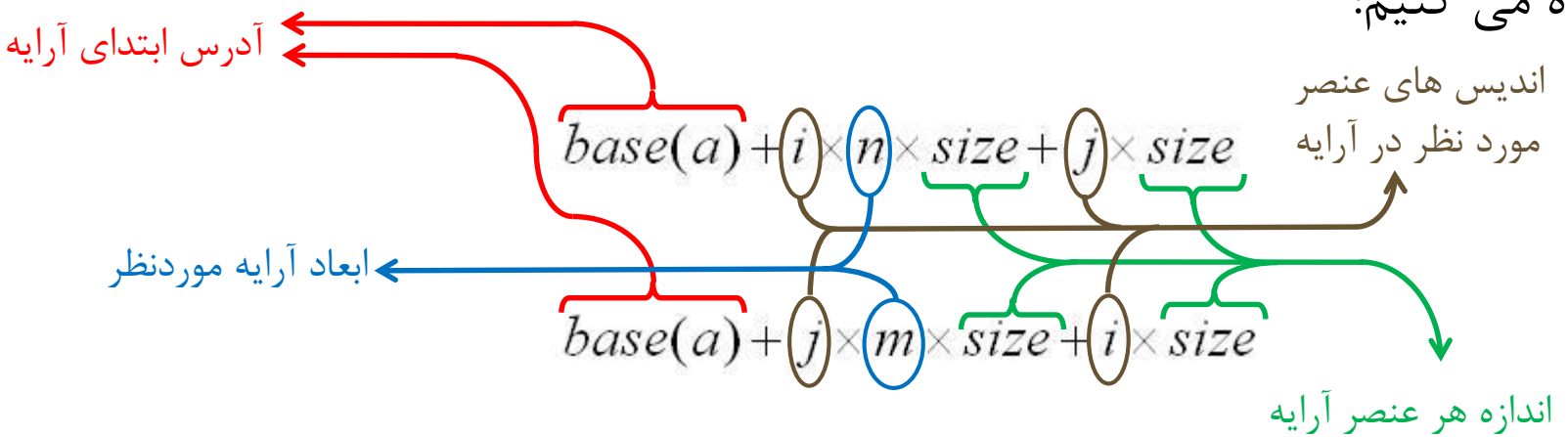
پیاده سازی آرایه دو بعدی

- دستور زیر به آسانی یک آرایه دو بعدی را پیاده سازی می کند :

```
int  a[m][n];
```

✓ این دستور $m \times n$ مکان متوالی حافظه را که هر کدام محل ذخیره سازی یک عدد صحیح است ، برای یک برنامه تخصیص می دهد.

- اگر بخواهیم آدرس فیزیکی محل ذخیره سازی داده مورد نظر را بیابیم، از رابطه های زیر استفاده می کنیم:



نکته! اندیس هر عنصر در آرایه یک واحد کمتر از شماره عنصر در آرایه است.

کاربردهای آرایه ها

- **آرایه های یک بعدی**
  - مرتب سازی لیست ها
  - جستجو در لیست ها

- **آرایه های دو بعدی**
  - جمع و تفریق ماتریس ها در ریاضیات
  - ضرب ماتریس ها در ریاضیات
  - پیاده سازی چند جمله ای ها
  - پیاده سازی ماتریس های خلوت (اِسپارس)

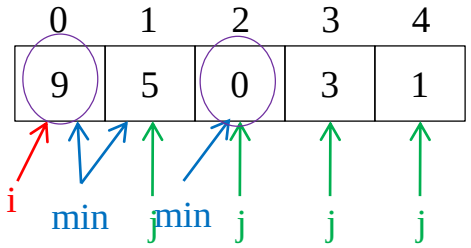
مرتب سازی لیست ها

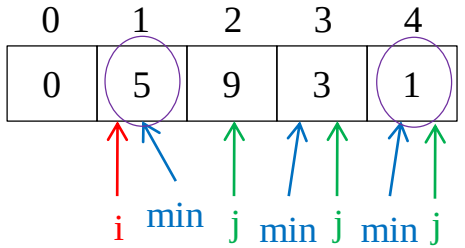
الگوریتم مرتب سازی انتخابی

دریافتی : مقدار صحیح $n >= 1$ و آرایه x

برگشتی : آرایه x که بصورت صعودی مرتب شده است.

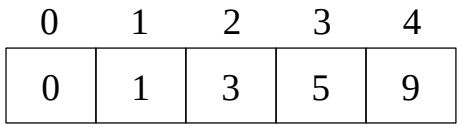
1. برای $i = 0$ تا $n-2$ مراحل زیر را انجام بده
   - 1-1. $min = i$
   - 1-2. برای $j = i + 1$ تا $n-1$ اعمال زیر را انجام بده
     - 1-2-1. اگر $x[j] < x[min]$ آنگاه $min = j$
   - 1-3. $item = x[i]$
   - 1-4. $x[i] = x[min]$
   - 1-5. $x[min] = item$

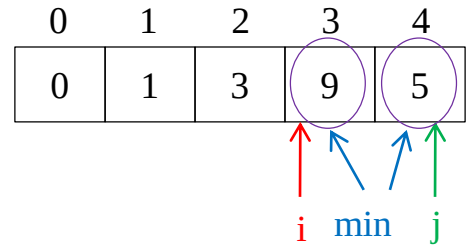


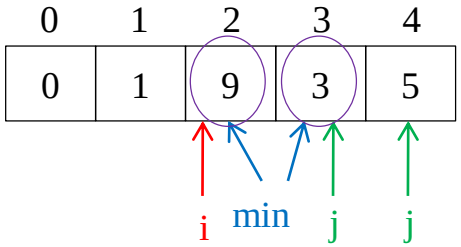


i	j	min	item
0	1	0	-
0	2	1	-
0	3	2	-
0	4	2	9

i	j	min	item
1	2	1	-
1	3	1	-
1	4	~~3~~ 4	5







i	j	min	item
3	4	~~3~~ 4	9

i	j	min	item
2	3	2	-
2	4	3	9

جستجو در لیست ها

الگوریتم جستجوی ترتیبی

دریافتی : مقدار صحیح n ، آرایه x به طول n و item که باید جستجو شود.

برگشتی : اگر item وجود داشته باشد، مقدار 1 و در غیر اینصورت مقدار صفر برگردانده می شود

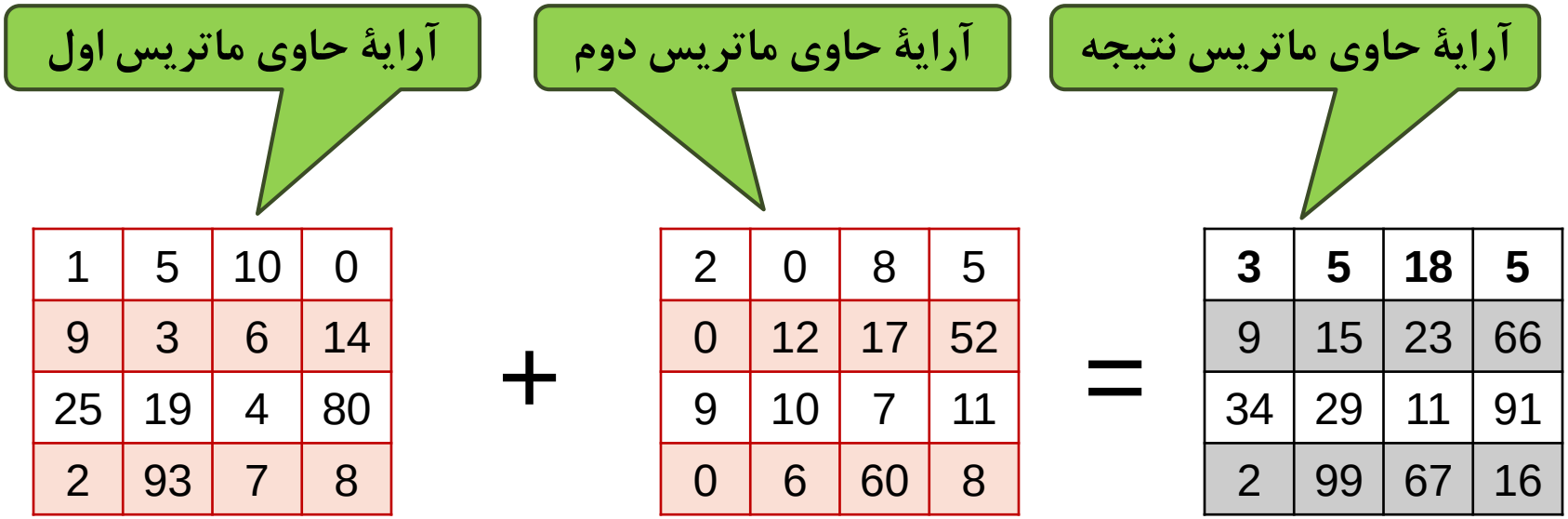
1. 0 را در found قرار بده
2. 0 را در loc قرار بده
3. تا زمانی که loc < n و found = 0 مراحل زیر را انجام بده
   - 3-1. اگر item == x[loc] آنگاه 1 را در found قرار بده

     در غیر اینصورت به loc یک واحد اضافه کن
4. found را برگردان

جمع و تفریق ماتریس ها

- برای جمع و تفریق دو ماتریس باید ابعاد دو ماتریس با هم برابر باشند.
- جمع و تفریق دو ماتریس بصورت نظیر به نظیر اعضای آنها انجام می شود.

مثال:



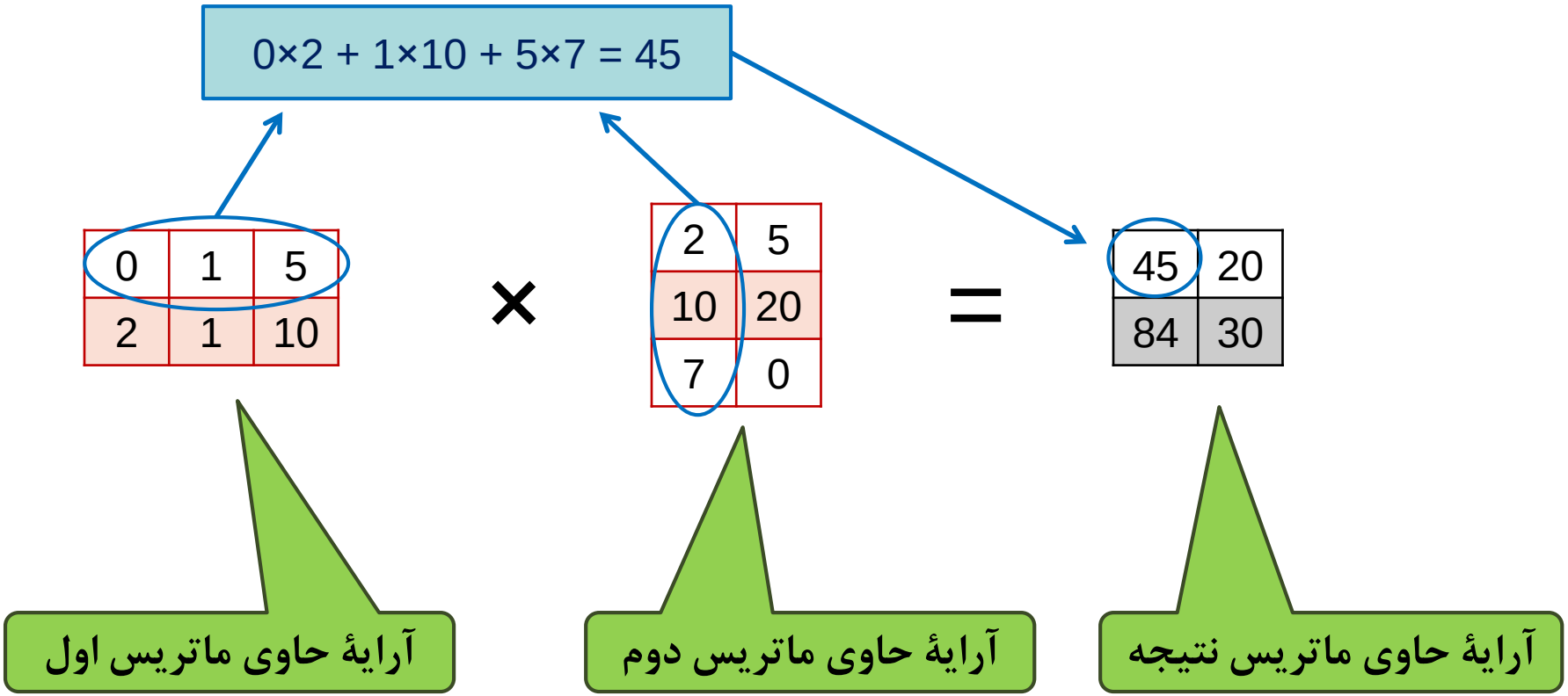
ضرب ماتریس ها

- برای ضرب دو ماتریس باید تعداد ستون های ماتریس اول با تعداد سطرهای ماتریس دوم با هم برابر باشند.
- ابعاد ماتریس حاصلضرب با سطرهای ماتریس اول و ستون های ماتریس برابر خواهد بود.

```
int i, j, k;
//multiply mat1 by mat2
for(i = 0 ; i < 2 ; i++)
    for(j = 0 ; j < 4 ; j++)
        for(k = 0 ;k < 3 ; k ++)
            m3[i][j] = m3[i][j] + m1[i][k]* m2[k][j];
```

ضرب ماتریس ها

مثال: ضرب ماتریس ها



پیاده سازی چند جمله ای ها

- **حالت اول: به کمک آرایه یک بعدی**
  - آرایه یک بعدی با حجم برابر با بزرگترین توان جملات چند جمله ای بعلاوه یک
  - اندیس های آرایه به عنوان توان جملات چند جمله ای
  - محتوای سلول های آرایه به عنوان ضرایب متناظر جملات چند جمله ای

مثال:

$$f(x) = 3x^6 + x^5 - 7x^4 + 5x - 4$$

$$f(x) = 3x^6 + x^5 - 7x^4 + 0x^3 + 0x^2 + 5x^1 - 4x^0$$

0	1	2	3	4	5	6
-4	5	0	0	-7	1	3

پیاده سازی چند جمله ای ها

- **حالت دوم: به کمک آرایه دو بعدی**
  - آرایه دو بعدی به اندازه ۲ سطر و تعداد ستون های برابر با جملات چند جمله ای
  - سطر اول آرایه توان جملات و سطر دوم آرایه ضرایب متناظر با جملات سطر اول

مثال:

$$f(x) = 3x^6 + x^5 - 7x^4 + 5x - 4$$

	0	1	2	3	4
0	6	5	4	1	0
1	3	1	-7	5	-4

پیاده سازی ماتریس خلوت

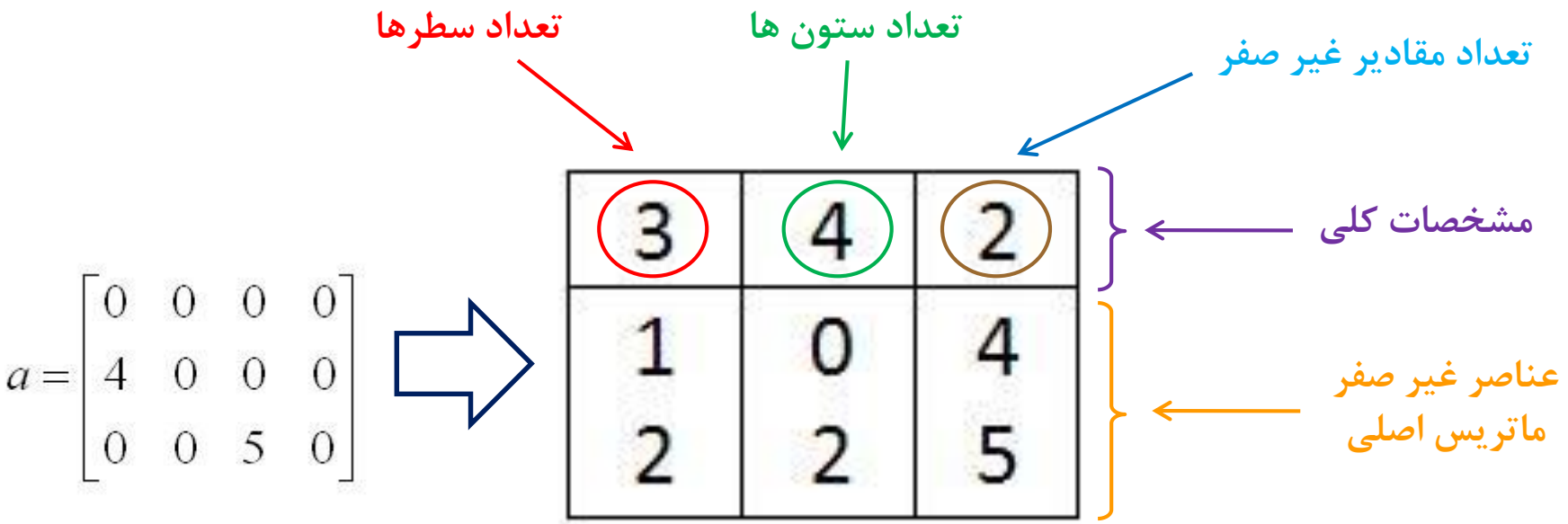
- **ماتریس های خلوت (اسپارس) (sparse)**
  - ماتریسی که تعداد زیادی از عناصر آن صفر باشد، ماتریس خلوت گفته می شود.

$$a = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 \end{bmatrix}$$

  - اعمال ریاضی که روی ماتریس های خلوت صورت می گیرد برای عناصر صفر بی اثر است.
  - ذخیره سازی عناصر صفر در حافظه عملاً مفید نیست زیرا تاثیری در محاسبات ندارند.
  - باید شیوه ای در ذخیره سازی مورد استفاده قرار گیرد تا فقط عناصر غیر صفر در حافظه ذخیره شوند.

پیاده سازی ماتریس خلوت

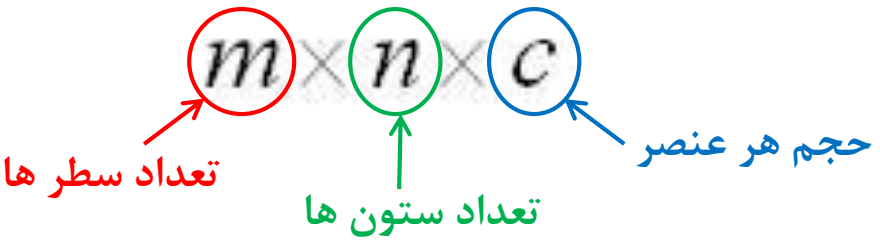
- روش ذخیره بهینه ماتریس خلوت



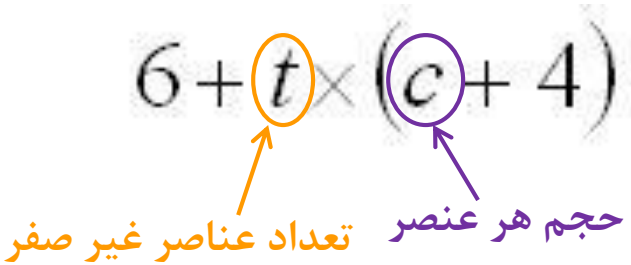
پیاده سازی ماتریس خلوت

- **مقایسه فضای ذخیره سازی**

۱. در حالت عادی :



۲. نمایش ماتریس خلوت :



❖ بر اساس روابط بالا زمانی از نمایش بهینه ماتریس خلوت استفاده می شود که نامساوی زیر برقرار باشد :

$$6 + t \times (c+4) < m \times n \times c \longrightarrow t < \frac{(m \times n \times c) - 6}{c+4}$$

مشکلات آرایه

- حد و مرز آرایه ها در زبان C کنترل نمی شود.
- ظرفیت آرایه در طول اجرای برنامه تغییر نمی کند.
- درج کردن یک مقدار جدید در آرایه مستلزم جابجایی همه عناصر یا برخی از عناصر آرایه است.
- حذف کردن عناصر از آرایه نیز مستلزم جابجایی عناصر آرایه است.