



اصول برنامه نویسی کامپیوتر

آرایه ها: فضای بیشتر برای داده ها

محمد فرشی

بخش علوم کامپیوتر - دانشکده علوم ریاضی - دانشگاه یزد

۲۱ دی ۱۴۰۳

فضای بیشتر برای داده‌ها

فضای بیشتر برای داده‌ها

در مسائلی چون محاسبهٔ مجموع، میانگین یا بیشترین مقدار n عدد، با تعداد زیادی داده روبرو بودیم، ولی نیازی به نگهداری همهٔ n عدد نداشتیم.

اما مسائلی هستند که در آنها تعداد زیادی داده دخیل هستند و نگهداری تمام آنها در حافظه برای استفاده از آنها در بخش‌های مختلف برنامه، لازم است.

اولین سؤالی که برای حل این مسئله مطرح می‌شود این است که این اعداد را کجا ذخیره کنیم؟

n متغیر تعریف کنیم: مشکلات: حجم بالای تعریف متغیر با نام‌های متفاوت و عدم مقیاس‌پذیری و کارکردن با آنها برای انجام کارها بسیار سخت است.

آرایه‌ها، سافت‌واری را در اختیار قرار می‌دهند که امکان ذخیره‌سازی تعداد زیادی داده از یک نوع را به سادگی فراهم می‌کنند.

استفاده از یک نام مشترک و یک (یا چند) اندیس عددی برای تمایز متغیرهای مختلف مورد استفاده قرار می‌گیرد.



فضای بیشتر برای داده‌ها

فضای بیشتر برای داده‌ها

در مسائلی چون محاسبهٔ مجموع، میانگین یا بیشترین مقدار n عدد، با تعداد زیادی داده روبرو بودیم، ولی نیازی به نگهداری همهٔ n عدد نداشتیم.

اما مسائلی هستند که در آنها تعداد زیادی داده دخیل هستند و نگهداری تمام آنها در حافظه برای استفاده از آنها در بخش‌های مختلف برنامه، لازم است.

اولین سؤالی که برای حل این مسئله مطرح می‌شود این است که این اعداد را کجا ذخیره کنیم؟
 n متغیر تعریف کنیم: مشکلات: حجم بالای تعریف متغیر با نام‌های متفاوت و عدم مقیاس‌پذیری و کارکردن با آنها برای انجام کارها بسیار سخت است.

آرایه‌ها، سافت‌واری را در اختیار قرار می‌دهند که امکان ذخیره‌سازی تعداد زیادی داده از یک نوع را به سادگی فراهم می‌کنند.

استفاده از یک نام مشترک و یک (یا چند) اندیس عددی برای تمایز متغیرهای مختلف مورد استفاده قرار می‌گیرد.



فضای بیشتر برای داده‌ها

فضای بیشتر برای داده‌ها

در مسائلی چون محاسبهٔ مجموع، میانگین یا بیشترین مقدار n عدد، با تعداد زیادی داده روبرو بودیم، ولی نیازی به نگهداری همهٔ n عدد نداشتیم.

اما مسائلی هستند که در آنها تعداد زیادی داده دخیل هستند و نگهداری تمام آنها در حافظه برای استفاده از آنها در بخش‌های مختلف برنامه، لازم است.

اولین سؤالی که برای حل این مسئله مطرح می‌شود این است که این اعداد را کجا ذخیره کنیم؟

n متغیر تعریف کنیم: مشکلات: حجم بالای تعریف متغیر با نام‌های متفاوت و عدم مقیاس‌پذیری و کارکردن با آنها برای انجام کارها بسیار سخت است.

آرایه‌ها، سافت‌آری را در اختیار قرار می‌دهند که امکان ذخیره‌سازی تعداد زیادی داده از یک نوع را به سادگی فراهم می‌کنند.

استفاده از یک نام مشترک و یک (یا چند) اندیس عددی برای تمایز متغیرهای مختلف مورد استفاده قرار می‌گیرد.



فضای بیشتر برای داده‌ها

فضای بیشتر برای داده‌ها

در مسائلی چون محاسبهٔ مجموع، میانگین یا بیشترین مقدار n عدد، با تعداد زیادی داده روبرو بودیم، ولی نیازی به نگهداری همهٔ n عدد نداشتیم.

اما مسائلی هستند که در آنها تعداد زیادی داده دخیل هستند و نگهداری تمام آنها در حافظه برای استفاده از آنها در بخش‌های مختلف برنامه، لازم است.

اولین سؤالی که برای حل این مسئله مطرح می‌شود این است که این اعداد را کجا ذخیره کنیم؟
 n متغیر تعریف کنیم: مشکلات: حجم بالای تعریف متغیر با نام‌های متفاوت و عدم مقیاس‌پذیری و کارکردن با آنها برای انجام کارها بسیار سخت است.

آرایه‌ها، سافت‌واری را در اختیار قرار می‌دهند که امکان ذخیره‌سازی تعداد زیادی داده از یک نوع را به سادگی فراهم می‌کنند.

استفاده از یک نام مشترک و یک (یا چند) اندیس عددی برای تمایز متغیرهای مختلف مورد استفاده قرار می‌گیرد.



فضای بیشتر برای داده‌ها

فضای بیشتر برای داده‌ها

در مسائلی چون محاسبهٔ مجموع، میانگین یا بیشترین مقدار n عدد، با تعداد زیادی داده روبرو بودیم، ولی نیازی به نگهداری همهٔ n عدد نداشتیم.

اما مسائلی هستند که در آنها تعداد زیادی داده دخیل هستند و نگهداری تمام آنها در حافظه برای استفاده از آنها در بخش‌های مختلف برنامه، لازم است.

اولین سؤال آنکه برای حل این مسئله مطرح می‌شود این است که این اعداد را کجا ذخیره کنیم؟
 n متغیر تعریف کنیم: مشکلات: حجم بالای تعریف متغیر با نام‌های متفاوت و عدم مقیاس‌پذیری و کارکردن با آنها برای انجام کارها بسیار سخت است.

آرایه‌ها، سافت‌آری را در اختیار قرار می‌دهند که امکان ذخیره‌سازی تعداد زیادی داده از یک نوع را به سادگی فراهم می‌کنند.

استفاده از یک نام مشترک و یک (یا چند) اندیس عددی برای تمایز متغیرهای مختلف مورد استفاده قرار می‌گیرد.



فضای بیشتر برای داده‌ها

فضای بیشتر برای داده‌ها

در مسائلی چون محاسبهٔ مجموع، میانگین یا بیشترین مقدار n عدد، با تعداد زیادی داده روبرو بودیم، ولی نیازی به نگهداری همهٔ n عدد نداشتیم.

اما مسائلی هستند که در آنها تعداد زیادی داده دخیل هستند و نگهداری تمام آنها در حافظه برای استفاده از آنها در بخش‌های مختلف برنامه، لازم است.

اولین سؤال آنکه برای حل این مسئله مطرح می‌شود این است که این اعداد را کجا ذخیره کنیم؟

n متغیر تعریف کنیم: مشکلات: حجم بالای تعریف متغیر با نام‌های متفاوت و عدم مقیاس‌پذیری و کارکردن با آنها برای انجام کارها بسیار سخت است.

آرایه‌ها، سافت‌آری را در اختیار قرار می‌دهند که امکان ذخیره‌سازی تعداد زیادی داده از یک نوع را به سادگی فراهم می‌کنند.

استفاده از یک نام مشترک و یک (یا چند) اندیس عددی برای تمایز متغیرهای مختلف مورد استفاده قرار می‌گیرد.

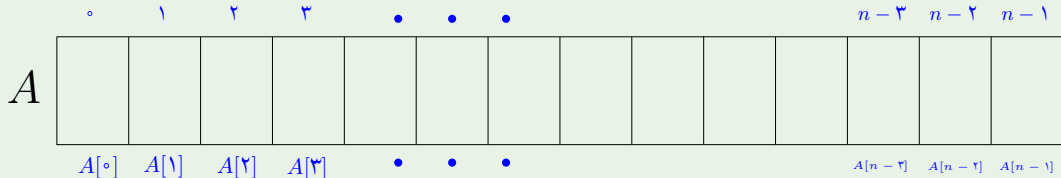


آرایهٔ یک بعدی

آرایهٔ یک بعدی

یک آرایه، از تعدادی متغیر **هم‌نوع** تشکیل شده است که در **فضایی به هم پیوسته** در حافظهٔ اصلی ذخیره می‌شوند و دارای ویژگی‌های زیر است.

هر آرایه دارای یک نام و یک طول است که طول آرایه همان تعداد خانه‌های (متغیرهای) آن آرایه است. به هر عنصر آرایه با طول n می‌توان به وسیلهٔ نام آرایه و یک اندیس بین صفر و $n - 1$ دسترسی داشت.

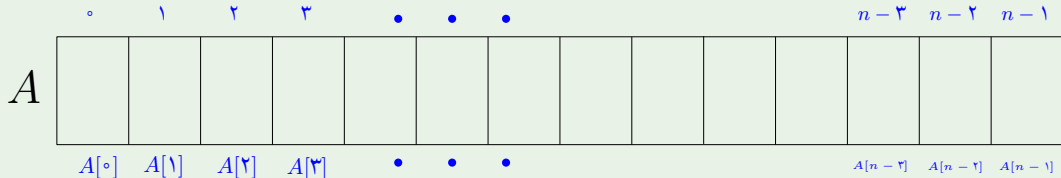


آرایهٔ یک بعدی

آرایهٔ یک بعدی

یک آرایه، از تعدادی متغیر **هم‌نوع** تشکیل شده است که در **فضایی به هم پیوسته** در حافظهٔ اصلی ذخیره می‌شوند و دارای ویژگی‌های زیر است.

هر آرایه دارای یک نام و یک طول است که طول آرایه همان تعداد خانه‌های (متغیرهای) آن آرایه است. به هر عنصر آرایه با طول n می‌توان به وسیلهٔ نام آرایه و یک اندیس بین **صفر** و $n - 1$ دسترسی داشت.



مثال ۱

گرفتن عدد n و سپس n عدد از ورودی و قرار دادن در آرایه A .



مثال ۱

گرفتن عدد n و سپس n عدد از ورودی و قرار دادن در آرایهٔ A .

الگوریتم:

- ۱ شروع
- ۲ n را بگیر
- ۳ $i \leftarrow 0$
- ۴ اگر $i < n$ آنگاه
- ۵ $A[i]$ را بگیر
- ۶ $i \leftarrow i + 1$
- ۷ به مرحلهٔ ۴ برو
- ۸ پایان اگر
- ۹ ...



مثال ۱

گرفتن عدد n و سپس n عدد از ورودی و قرار دادن در آرایهٔ A .

الگوریتم:

- ۱ شروع
- ۲ n را بگیر
- ۳ $i \leftarrow 0$
- ۴ اگر $i < n$ آنگاه
- ۵ $A[i]$ را بگیر
- ۶ $i \leftarrow i + 1$
- ۷ به مرحلهٔ ۴ برو
- ۸ پایان اگر
- ۹ ...

مقیاس‌پذیری برنامه: اجرای برنامه برای داده‌های با مقیاس مختلف، بدون نیاز به تغییر برنامه.



مثال ۲

الگوریتمی بنویسید که ابتدا عدد n و سپس n عدد را بگیرد و آن‌ها را به ترتیب عکس داده شده، چاپ کند.



مثال ۲

الگوریتمی بنویسید که ابتدا عدد n و سپس n عدد را بگیرد و آن‌ها را به ترتیب عکس داده شده، چاپ کند.

الگوریتم:

۱	شروع	۹	اگر $i > ۰$ آنگاه
۲	n را بگیر	۱۰	$i \leftarrow i - ۱$
۳	$i \leftarrow ۰$	۱۱	$A[i]$ را چاپ کن
۴	اگر $i < n$ آنگاه	۱۲	به مرحلهٔ ۹ برو
۵	$A[i]$ را بگیر	۱۳	پایان اگر
۶	$i \leftarrow i + ۱$	۱۴	پایان
۷	به مرحلهٔ ۴ برو	۱۵	
۸	پایان اگر		

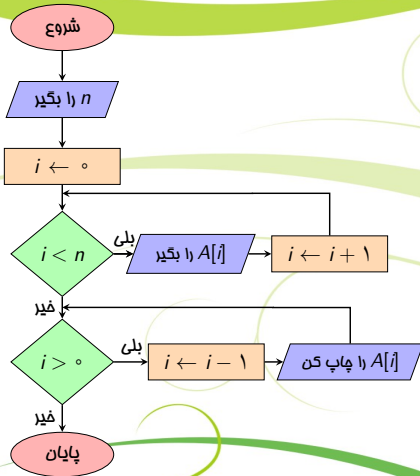


مثال ۲

الگوریتمی بنویسید که ابتدا عدد n و سپس n عدد را بگیرد و آن‌ها را به ترتیب عکس داده شده، چاپ کند.

الگوریتم:

۱	شروع	۹	اگر $i > 0$ آنگاه
۲	n را بگیر	۱۰	$i \leftarrow i - 1$
۳	$i \leftarrow 0$	۱۱	$A[i]$ را چاپ کن
۴	اگر $i < n$ آنگاه	۱۲	به مرحله ۹ برو
۵	$A[i]$ را بگیر	۱۳	پایان اگر
۶	$i \leftarrow i + 1$	۱۴	پایان
۷	به مرحله ۴ برو	۱۵	
۸	پایان اگر		



برنامه:

```

1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int i,n;
6     float A[200];
7     cout<<"Enter An Integer
8         (<=200):";
9     cin>>n;
10    for (i=0; i<n;i++)
11    {
12        cout<<"\n Enter Next
13            number:";
14        cin>>A[i];
15    }
16    while (i>0)
17    {
18        i--;
19        cout<<A[i]<<" ";
20    }
21    return 0;

```

مثال ۲

الگوریتمی بنویسید که ابتدا عدد n و سپس n عدد را بگیرد و آن‌ها را به ترتیب عکس داده شده، چاپ کند.

الگوریتم:

۱	شروع	۹	اگر $i > 0$ آنگاه
۲	n را بگیر	۱۰	$i \leftarrow i - 1$
۳	$i \leftarrow 0$	۱۱	$A[i]$ را چاپ کن
۴	اگر $i < n$ آنگاه	۱۲	به مرحله ۹ برو
۵	$A[i]$ را بگیر	۱۳	پایان اگر
۶	$i \leftarrow i + 1$	۱۴	پایان
۷	به مرحله ۴ برو	۱۵	
۸	پایان اگر		

مثال ۳

الگوریتم و فلوچارتی ارائه کنید که ابتدا n و سپس n نمره را دریافت کرده و تعداد نمرات بالاتر از میانگین کلاس را چاپ کند.



مثال ۳

الگوریتم و فلوچارتی ارائه کنید که ابتدا n و سپس n نمره را دریافت کرده و تعداد نمرات بالاتر از میانگین کلاس را چاپ کند.

الگوریتم:

۱	شروع	۱۲	اگر $i < n$ آنگاه
۲	n را بگیر	۱۳	اگر $A[i] > Ave$ آنگاه
۳	$i \leftarrow i + 1$, $Sum \leftarrow 0$	۱۴	$counter \leftarrow counter + 1$
۴	اگر $i < n$ آنگاه	۱۵	پایان اگر
۵	$A[i]$ را بگیر	۱۶	$i \leftarrow i + 1$
۶	$Sum \leftarrow Sum + A[i]$	۱۷	به مرحله ۱۲ برو
۷	$i \leftarrow i + 1$	۱۸	پایان اگر
۸	به مرحله ۴ برو	۱۹	$counter$ را چاپ کن
۹	پایان اگر	۲۰	پایان
۱۰	$Ave \leftarrow Sum/n$		
۱۱	$i \leftarrow 0$, $counter \leftarrow 0$		

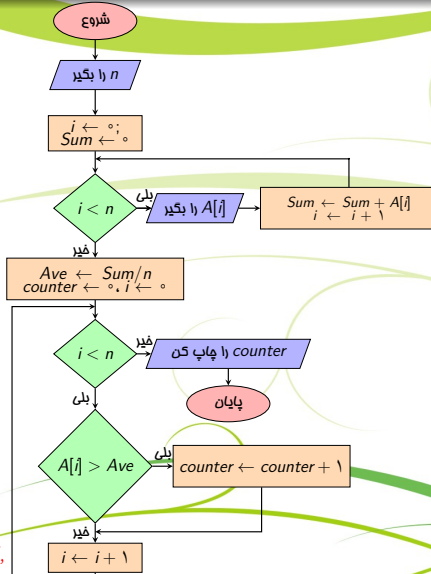


مثال ۳

الگوریتم و فلوچارتی ارائه کنید که ابتدا n و سپس n نمره را دریافت کرده و تعداد نمرات بالاتر از میانگین کلاس را چاپ کند.

الگوریتم:

۱	شروع	۱۲	اگر $i < n$ آنگاه
۲	n را بگیر	۱۳	اگر $A[i] > Ave$ آنگاه
۳	$i \leftarrow 0, Sum \leftarrow 0$	۱۴	$counter \leftarrow counter + 1$
۴	اگر $i < n$ آنگاه	۱۵	پایان اگر
۵	$A[i]$ را بگیر	۱۶	$i \leftarrow i + 1$
۶	$Sum \leftarrow Sum + A[i]$	۱۷	به مرحله ۱۲ برو
۷	$i \leftarrow i + 1$	۱۸	پایان اگر
۸	به مرحله ۴ برو	۱۹	$counter$ را چاپ کن
۹	پایان اگر	۲۰	پایان
۱۰	$Ave \leftarrow Sum/n$		
۱۱	$i \leftarrow 0, counter \leftarrow 0$		



برنامه:

```

1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int i, n, counter;
6     float A[200], Sum, Ave;
7     cout<<"Enter An Integer
8         (<=200).:";
9     cin>>n;
10    for (i=0; i<n;i++)
11    {
12        cout<<"\n Enter Next
13            number.:";
14        cin>>A[i];
15        Sum+=A[i];
16    }
17    Ave=Sum/n;
18    counter=0;
19    for (i=0; i<n; i++)
20    {
21        if (A[i]>Ave)
22            counter++;
23    }
24    cout<<"# of grades above
25        average.:"<<counter;
26    return 0;

```

مثال ۳

الگوریتم و فلوچارتی ارائه کنید که ابتدا n و سپس n نمره را دریافت کرده و تعداد نمرات بالاتر از میانگین کلاس را چاپ کند.

الگوریتم:

۱	شروع	۱۲	اگر $i < n$ آنگاه
۲	n را بگیر	۱۳	اگر $A[i] > Ave$ آنگاه
۳	$i \leftarrow i + 1$, $Sum \leftarrow Sum + A[i]$	۱۴	$counter \leftarrow counter + 1$
۴	اگر $i < n$ آنگاه	۱۵	پایان اگر
۵	$A[i]$ را بگیر	۱۶	$i \leftarrow i + 1$
۶	$Sum \leftarrow Sum + A[i]$	۱۷	به مرحله ۱۲ برو
۷	$i \leftarrow i + 1$	۱۸	پایان اگر
۸	به مرحله ۴ برو	۱۹	$counter$ را چاپ کن
۹	پایان اگر	۲۰	پایان
۱۰	$Ave \leftarrow Sum/n$		
۱۱	$i \leftarrow i + 1$, $counter \leftarrow counter + 1$		

مثال ۴

زیرالگوریتمی بنویسید که یک آرایه از اعداد و اندازه آن را دریافت کرده و عناصر موجود در آرایه را به ترتیب صعودی مرتب کند.



مثال ۴

زیرالگوریتمی بنویسید که یک آرایه از اعداد و اندازهٔ آن را دریافت کرده و عناصر موجود در آرایه را به ترتیب صعودی مرتب کند.

A

۱۰	۷۵	۵	۹	۱	۱۲۵
----	----	---	---	---	-----



مثال ۴

زیرالگوریتمی بنویسید که یک آرایه از اعداد و اندازه آن را دریافت کرده و عناصر موجود در آرایه را به ترتیب صعودی مرتب کند.

الگوریتم:

۱ شروع $Sort(A, n)$	۱۶ پایان اگر
۲ $i \leftarrow 0$	۱۷ A را برگردان
۳ اگر $i < n$ آنگاه	
۴ $j \leftarrow i + 1$	
۵ اگر $j < n$ آنگاه	
۶ اگر $A[i] > A[j]$ آنگاه	
۷ $B \leftarrow A[i]$	
۸ $A[i] \leftarrow A[j]$	
۹ $A[j] \leftarrow B$	
۱۰ پایان اگر	
۱۱ $j \leftarrow j + 1$	
۱۲ به مرحله ۵ برو	
۱۳ پایان اگر	
۱۴ $i \leftarrow i + 1$	
۱۵ به مرحله ۳ برو	

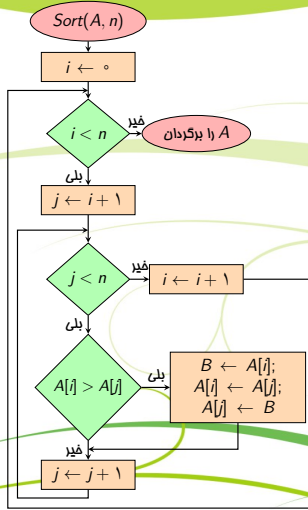


مثال ۴

زیرالگوریتمی بنویسید که یک آرایه از اعداد و اندازهٔ آن را دریافت کرده و عناصر موجود در آرایه را به ترتیب صعودی مرتب کند.

الگوریتم:

۱	شروع $Sort(A, n)$	۱۶	پایان اگر
۲	$i \leftarrow 0$	۱۷	A را برگردان
۳	اگر $i < n$ آنگاه		
۴	$j \leftarrow i + 1$		
۵	اگر $j < n$ آنگاه		
۶	اگر $A[i] > A[j]$ آنگاه		
۷	$B \leftarrow A[j]$		
۸	$A[i] \leftarrow A[j]$		
۹	$A[j] \leftarrow B$		
۱۰	پایان اگر		
۱۱	$j \leftarrow j + 1$		
۱۲	به مرحلهٔ ۵ برو		
۱۳	پایان اگر		
۱۴	$i \leftarrow i + 1$		
۱۵	به مرحلهٔ ۳ برو		



مثال ۴

زیرالگوریتمی بنویسید که یک آرایه از اعداد و اندازهٔ آن را دریافت کرده و عناصر موجود در آرایه را به ترتیب صعودی مرتب کند.

الگوریتم:

۱	شروع $Sort(A, n)$	۱۶	پایان اگر
۲	$i \leftarrow 0$	۱۷	A را برگردان
۳	اگر $i < n$ آنگاه		
۴	$j \leftarrow i + 1$		
۵	اگر $j < n$ آنگاه		
۶	اگر $A[i] > A[j]$ آنگاه		
۷	$B \leftarrow A[j]$		
۸	$A[j] \leftarrow A[i]$		
۹	$A[i] \leftarrow B$		
۱۰	پایان اگر		
۱۱	$j \leftarrow j + 1$		
۱۲	به مرحلهٔ ۵ برو		
۱۳	پایان اگر		
۱۴	$i \leftarrow i + 1$		
۱۵	به مرحلهٔ ۳ برو		

برنامه:

```

1 void Sort(double A[500], int n)
2 {
3     int i, j;
4     double B;
5     for (i=0; i<n; i++)
6         for (j=i+1; j<n; j++)
7             if (A[i]>A[j])
8                 {
9                     B=A[j];
10                    A[j]=A[i];
11                    A[i]=B;
12                }
13     return ;
14 }
```

مثال ۵

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n عدد را بگیرد و آنها را به ترتیب صعودی مرتب کرده و چاپ کند.



مثال ۵

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n عدد را بگیرد و آنها را به ترتیب صعودی مرتب کرده و چاپ کند.

الگوریتم:

۱ شروع $Input_Array(A, n)$

۲ $i \leftarrow 0$

۳ اگر $i < n$ آنگاه

۴ $A[i]$ را بگیر

۵ $i \leftarrow i + 1$

۶ به مرحله ۳ برو

۷ پایان اگر

۸ A را برگردان



مثال ۵

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n عدد را بگیرد و آنها را به ترتیب صعودی مرتب کرده و چاپ کند.

الگوریتم:

۱ شروع $Input_Array(A, n)$

۲ $i \leftarrow 0$

۳ اگر $i < n$ آنگاه

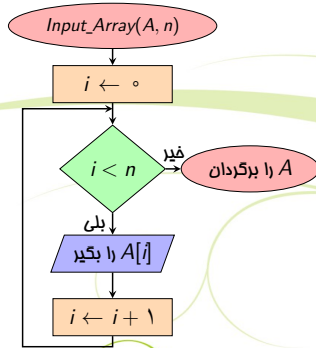
۴ $A[i]$ را بگیر

۵ $i \leftarrow i + 1$

۶ به مرحله ۳ برو

۷ پایان اگر

۸ A را برگردان



مثال ۵

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n عدد را بگیرد و آنها را به ترتیب صعودی مرتب کرده و چاپ کند.

الگوریتم:

۱ شروع $Input_Array(A, n)$

۲ $i \leftarrow 0$

۳ اگر $i < n$ آنگاه

۴ $A[i]$ را بگیر

۵ $i \leftarrow i + 1$

۶ به مرحله ۳ برو

۷ پایان اگر

۸ A را برگردان

برنامه:

```
1 void Input_Array(double A[500], int n)
2 {
3     int i;
4     for(i=0; i<n; i++)
5     {
6         cout<<"Enter Next Element:";
7         cin>>A[i];
8     }
9     return;
10 }
```



مثال ۶

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n عدد را بگیرد و آنها را به ترتیب صعودی مرتب کرده و چاپ کند.



مثال ۶

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n عدد را بگیرد و آنها را به ترتیب صعودی مرتب کرده و چاپ کند.

الگوریتم:

۱ شروع $Output_Array(A, n)$ ۲ $i \leftarrow 0$ ۳ اگر $i < n$ آنگاه۴ $A[i]$ را چاپ کن۵ $i \leftarrow i + 1$

۶ به مرحله ۳ برو

۷ پایان اگر

۸ برگرد



مثال ۶

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n عدد را بگیرد و آنها را به ترتیب صعودی مرتب کرده و چاپ کند.

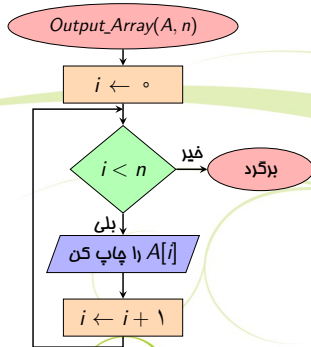
الگوریتم:

۱ شروع $Output_Array(A, n)$ ۲ $i \leftarrow 0$ ۳ اگر $i < n$ آنگاه۴ $A[i]$ را چاپ کن۵ $i \leftarrow i + 1$

۶ به مرحله ۳ برو

۷ پایان اگر

۸ برگرد



مثال ۶

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n عدد را بگیرد و آنها را به ترتیب صعودی مرتب کرده و چاپ کند.

برنامه:

```
1 void Output_Array(double A[500], int n)
2 {
3     int i;
4     for(i=0; i<n; i++)
5         cout<<A[i]<<" ";
6     return;
7 }
```

الگوریتم:

- ۱ شروع $Output_Array(A, n)$
- ۲ $i \leftarrow 0$
- ۳ اگر $i < n$ آنگاه
- ۴ $A[i]$ را چاپ کن
- ۵ $i \leftarrow i + 1$
- ۶ به مرحله ۳ برو
- ۷ پایان اگر
- ۸ برگرد



مثال ۷

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n عدد را بگیرد و آنها را به ترتیب صعودی مرتب کرده و چاپ کند.



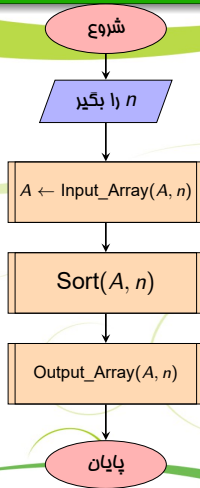
مثال ۷

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n عدد را بگیرد و آنها را به ترتیب صعودی مرتب کرده و چاپ کند.

الگوریتم:

- ۱ شروع
- ۲ n را بگیر
- ۳ $A \leftarrow \text{Input_Array}(A, n)$
- ۴ $\text{Sort}(A, n)$
- ۵ $\text{Output_Array}(A, n)$
- ۶ پایان





مثال ۷

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n عدد را بگیرد و آنها را به ترتیب صعودی مرتب کرده و چاپ کند.

الگوریتم:

- ۱ شروع
- ۲ n را بگیر
- ۳ $A \leftarrow \text{Input_Array}(A, n)$
- ۴ $\text{Sort}(A, n)$
- ۵ $\text{Output_Array}(A, n)$
- ۶ پایان



برنامه:

```
1 int main()  
2 {  
3     int n;  
4     double A[500];  
5     cout << "Enter n: ";  
6     cin >> n;  
7     Input_Array(A,n);  
8     Sort(A,n);  
9     Output_Array(A,n);  
10    return 0;  
11 }
```

مثال ۷

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n عدد را بگیرد و آنها را به ترتیب صعودی مرتب کرده و چاپ کند.

الگوریتم:

- ۱ شروع
- ۲ n را بگیر
- ۳ $A \leftarrow \text{Input_Array}(A, n)$
- ۴ $\text{Sort}(A, n)$
- ۵ $\text{Output_Array}(A, n)$
- ۶ پایان



مثال ۸

جستجوی داده‌ها: آیا عدد x در آرایه A شامل n عدد موجود است؟



مثال ۸

جستجوی داده‌ها: آیا عدد x در آرایهٔ A شامل n عدد موجود است؟

الگوریتم:

۱ شروع $Search(A, n, x)$

۲ $i \leftarrow 0$

۳ اگر $i < n$ آنگاه

۴ اگر $x = A[i]$ آنگاه

۵ ۱ را برگردان

۶ پایان اگر

۷ $i \leftarrow i + 1$

۸ به مرحلهٔ ۳ برو

۹ پایان اگر

۱۰ ۰ را برگردان

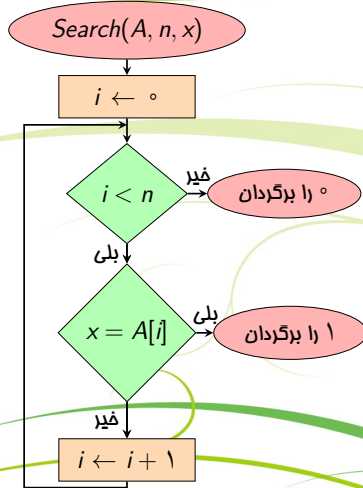


مثال ۸

جستجوی داده‌ها: آیا عدد x در آرایه A شامل n عدد موجود است؟

الگوریتم:

- ۱ شروع $Search(A, n, x)$
- ۲ $i \leftarrow 0$
- ۳ اگر $i < n$ آنگاه
- ۴ اگر $x = A[i]$ آنگاه
- ۵ ۱ را برگردان
- ۶ پایان اگر
- ۷ $i \leftarrow i + 1$
- ۸ به مرحله ۳ برو
- ۹ پایان اگر
- ۱۰ ۰ را برگردان



مثال ۸

جستجوی داده‌ها: آیا عدد x در آرایه A شامل n عدد موجود است؟

الگوریتم:

- ۱ شروع $Search(A, n, x)$
- ۲ $i \leftarrow 0$
- ۳ اگر $i < n$ آنگاه
- ۴ اگر $x = A[i]$ آنگاه
- ۵ i را برگردان
- ۶ پایان اگر
- ۷ $i \leftarrow i + 1$
- ۸ به مرحله ۳ برو
- ۹ پایان اگر
- ۱۰ i را برگردان

برنامه:

```

1 int Search(double A[ ],
2             int n, double x)
3 {
4     int i=0;
5     for (i=0; i<n; i++)
6     {
7         if (x == A[i])
8             return 1;
9     }
10    return 0;
11 }
```

مثال ۹

جستجوی داده ها: آیا عدد x در آرایه A شامل n عدد موجود است؟



مثال ۹

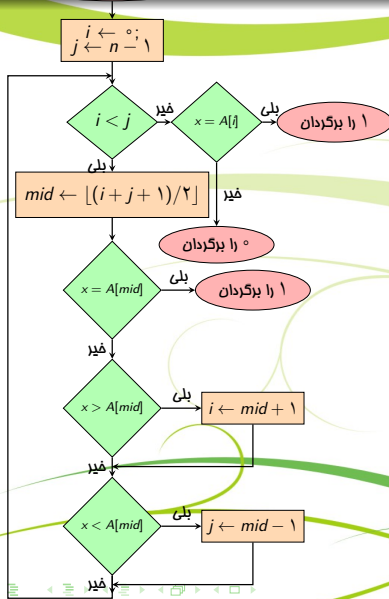
جستجوی داده‌ها: آیا عدد x در آرایه A شامل n عدد موجود است؟

الگوریتم:

- ۱ شروع $Bin_Search(A, n, x)$ پایان اگر
- ۲ $j \leftarrow n - 1; i \leftarrow 0$ اگر $x = A[i]$ آنگاه
- ۳ اگر $j < i$ آنگاه
- ۴ $mid \leftarrow \lfloor (i + j + 1) / 2 \rfloor$
- ۵ اگر $x = A[mid]$ آنگاه
- ۶ $i \leftarrow mid + 1$ یا برگردان
- ۷ پایان اگر
- ۸ اگر $x > A[mid]$ آنگاه
- ۹ $i \leftarrow mid + 1$
- ۱۰ پایان اگر
- ۱۱ اگر $x < A[mid]$ آنگاه
- ۱۲ $j \leftarrow mid - 1$
- ۱۳ پایان اگر
- ۱۴ به مرحله ۳ برو



$Bin_Search(A, n, x)$



مثال ۹

جستجوی داده‌ها: آیا عدد x در آرایه A شامل n عدد موجود است؟

الگوریتم:

- ۱ شروع $Bin_Search(A, n, x)$ پایان اگر
- ۲ $j \leftarrow n - 1; i \leftarrow 0$ اگر $x = A[i]$ آنگاه
- ۳ اگر $i < j$ آنگاه
- ۴ $mid \leftarrow \lfloor (i + j + 1) / 2 \rfloor$
- ۵ اگر $x = A[mid]$ آنگاه
- ۶ ۱ را برگردان
- ۷ پایان اگر
- ۸ اگر $x > A[mid]$ آنگاه
- ۹ $i \leftarrow mid + 1$
- ۱۰ پایان اگر
- ۱۱ اگر $x < A[mid]$ آنگاه
- ۱۲ $j \leftarrow mid - 1$
- ۱۳ پایان اگر
- ۱۴ به مرحله ۳ برو



مثال ۹

جستجوی داده‌ها: آیا عدد x در آرایه A شامل n عدد موجود است؟

۰	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳
۲	۱۱	۲۰	۳۵	۳۶	۸۱	۱۰۲	۲۱۲	۳۴۲	۴۲۰	۵۵۵	۵۸۰	۶۵۷	۷۲۱

n	x	i	j	mid
۱۴	۱۰۲	۰	۱۳	۷
			۶	۳
		۴		۵
		۶		

الگوریتم:

- ۱ شروع $Bin_Search(A, n, x)$ پایان اگر
- ۲ $i \leftarrow 0; j \leftarrow n - 1$ اگر $x = A[i]$ آنگاه
- ۳ اگر $j < i$ آنگاه
- ۴ $mid \leftarrow \lfloor (i + j + 1) / 2 \rfloor$
- ۵ اگر $x = A[mid]$ آنگاه
- ۶ $i \leftarrow mid + 1$ را برگردان
- ۷ پایان اگر
- ۸ اگر $x > A[mid]$ آنگاه
- ۹ $i \leftarrow mid + 1$
- ۱۰ پایان اگر
- ۱۱ اگر $x < A[mid]$ آنگاه
- ۱۲ $j \leftarrow mid - 1$
- ۱۳ پایان اگر
- ۱۴ به مرحله ۳ برو



برنامه:

```

1 int Bin_Search(double A[],
2               int n, double x)
3 {
4     int i=0, j=n-1, mid;
5     while(i < j)
6     {
7         mid=(i+j+1)/2;
8         if (x == A[mid])
9             return 1;
10        if (x > A[mid])
11            i=mid+1;
12        if (x < A[mid])
13            j=mid-1;
14    }
15    if (x == A[i])
16        return 1;
17    else
18        return 0;
19 }

```

مثال ۹

جستجوی داده‌ها: آیا عدد x در آرایه A شامل n عدد موجود است؟

الگوریتم:

۱	شروع $Bin_Search(A, n, x)$	۱۵	پایان اگر
۲	$i \leftarrow 0; j \leftarrow n - 1$	۱۶	اگر $x = A[i]$ آنگاه
۳	اگر $j < i$ آنگاه	۱۷	۱ را برگردان
۴	$mid \leftarrow \lfloor (i + j + 1) / 2 \rfloor$	۱۸	در غیر این صورت
۵	اگر $x = A[mid]$ آنگاه	۱۹	۰ را برگردان
۶	۱ را برگردان	۲۰	پایان اگر
۷	پایان اگر		
۸	اگر $x > A[mid]$ آنگاه		
۹	$i \leftarrow mid + 1$		
۱۰	پایان اگر		
۱۱	اگر $x < A[mid]$ آنگاه		
۱۲	$j \leftarrow mid - 1$		
۱۳	پایان اگر		
۱۴	به مرحله ۳ برو		

مثال ۱۰

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را مناسبه و چاپ کند.



مثال ۱۰

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را محاسبه و چاپ کند.

سه زیرمسئله:

زیرمسئله گرفتن n عدد و قرار دادن در آرایه.

زیرمسئله محاسبه بیشترین عدد در آرایه.

زیرمسئله محاسبه تعداد تکرار یک عدد در آرایه.



مثال ۱۰

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را محاسبه و چاپ کند.

سه زیرمسئله:

زیرمسئلهٔ گرفتن n عدد و قرار دادن در آرایه.

زیرمسئلهٔ محاسبهٔ بزرگ‌ترین عدد در آرایه.

زیرمسئلهٔ محاسبهٔ تعداد تکرار یک عدد در آرایه.



مثال ۱۰

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را محاسبه و چاپ کند.

سه زیرمسئله:

زیرمسئلهٔ گرفتن n عدد و قرار دادن در آرایه.

زیرمسئلهٔ محاسبهٔ بزرگ‌ترین عدد در آرایه.

زیرمسئلهٔ محاسبهٔ تعداد تکرار یک عدد در آرایه.



مثال ۱۰

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را محاسبه و چاپ کند.

سه زیرمسئله:

زیرمسئله گرفتن n عدد و قرار دادن در آرایه.

زیرمسئله محاسبه بزرگ‌ترین عدد در آرایه.

زیرمسئله محاسبه تعداد تکرار یک عدد در آرایه.

الگوریتم:

```

۱ شروع  $Max\_Element(A, n)$ 
۲  $Max \leftarrow A[0]$ 
۳  $i \leftarrow 1$ 
۴ اگر  $i < n$  آنگاه
۵   اگر  $Max < A[i]$  آنگاه
۶      $Max \leftarrow A[i]$ 
۷   پایان اگر
۸    $i \leftarrow i + 1$ 
۹ به مرحله ۳ برو
۱۰ پایان اگر
۱۱  $Max$  را برگردان
    
```



مثال ۱۰

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را محاسبه و چاپ کند.

سه زیرمسئله:

زیرمسئله گرفتن n عدد و قرار دادن در آرایه.

زیرمسئله محاسبه بزرگ‌ترین عدد در آرایه.

زیرمسئله محاسبه تعداد تکرار یک عدد در آرایه.

الگوریتم:

۱ شروع $Max_Element(A, n)$

۲ $Max \leftarrow A[0]$

۳ $i \leftarrow 1$

۴ اگر $i < n$ آنگاه

۵ اگر $Max < A[i]$ آنگاه

۶ $Max \leftarrow A[i]$

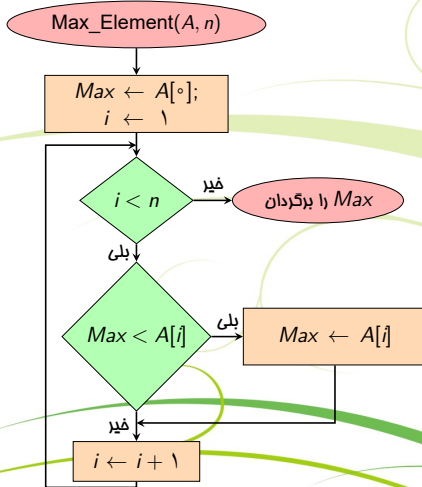
۷ پایان اگر

۸ $i \leftarrow i + 1$

۹ به مرحله ۳ برو

۱۰ پایان اگر

۱۱ Max را برگردان



مثال ۱۰

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را محاسبه و چاپ کند.

سه زیرمسئله:

زیرمسئله گرفتن n عدد و قرار دادن در آرایه.

زیرمسئله محاسبه بزرگ‌ترین عدد در آرایه.

زیرمسئله محاسبه تعداد تکرار یک عدد در آرایه.

الگوریتم:

۱	شروع $Max_Element(A, n)$
۲	$Max \leftarrow A[0]$
۳	$i \leftarrow 1$
۴	اگر $i < n$ آنگاه
۵	اگر $Max < A[i]$ آنگاه
۶	$Max \leftarrow A[i]$
۷	پایان اگر
۸	$i \leftarrow i + 1$
۹	به مرحله ۳ برو
۱۰	پایان اگر
۱۱	Max را برگردان

برنامه:

```

1 double Max_Element(double A[],
2                       int n)
3 {
4     double Max=A[0];
5     int i;
6     for(i=1; i<n; i++)
7         if (Max<A[i])
8             Max=A[i];
9     return Max;
10 }
```



مثال ۱۰

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را محاسبه و چاپ کند.

سه زیرمسئله:

زیرمسئله گرفتن n عدد و قرار دادن در آرایه.

زیرمسئله محاسبهٔ بزرگ‌ترین عدد در آرایه.

زیرمسئله محاسبهٔ تعداد تکرار یک عدد در آرایه.



مثال ۱۰

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را محاسبه و چاپ کند.

سه زیرمسئله:

زیرمسئله گرفتن n عدد و قرار دادن در آرایه.

زیرمسئله محاسبه بزرگ‌ترین عدد در آرایه.

زیرمسئله محاسبه تعداد تکرار یک عدد در آرایه.

الگوریتم:

```

۱ شروع  $Count(A, n, x)$ 
۲  $counter \leftarrow 0$ 
۳ اگر  $i < n$  آنگاه
۴   اگر  $A[i] = x$  آنگاه
۵      $counter \leftarrow counter + 1$ 
۶   پایان اگر
۷    $i \leftarrow i + 1$ 
۸ به مرحله ۳ برو
۹ پایان اگر
۱۰  $counter$  را برگردان
    
```



مثال ۱۰

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را محاسبه و چاپ کند.

سه زیرمسئله:

زیرمسئله گرفتن n عدد و قرار دادن در آرایه.

زیرمسئله محاسبه بزرگ‌ترین عدد در آرایه.

زیرمسئله محاسبه تعداد تکرار یک عدد در آرایه.

الگوریتم:

۱ شروع $Count(A, n, x)$

۲ $counter \leftarrow 0; i \leftarrow 0$

۳ اگر $i < n$ آنگاه

۴ اگر $A[i] = x$ آنگاه

۵ $counter \leftarrow counter + 1$

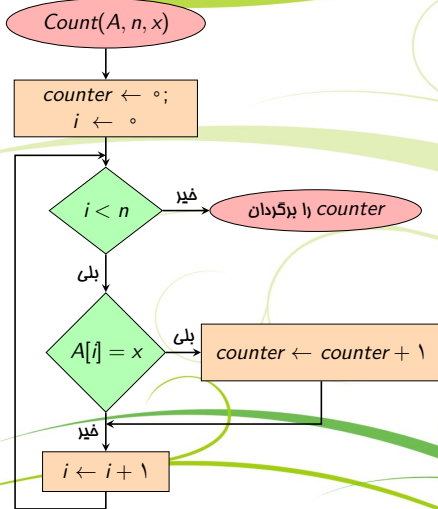
۶ پایان اگر

۷ $i \leftarrow i + 1$

۸ به مرحله ۳ برو

۹ پایان اگر

۱۰ $counter$ را برگردان



مثال ۱۰

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را محاسبه و چاپ کند.

سه زیرمسئله:

زیرمسئله گرفتن n عدد و قرار دادن در آرایه.

زیرمسئله محاسبه بزرگ‌ترین عدد در آرایه.

زیرمسئله محاسبه تعداد تکرار یک عدد در آرایه.

الگوریتم:

۱	شروع $Count(A, n, x)$
۲	$counter \leftarrow 0$ ؛ $i \leftarrow 0$
۳	اگر $i < n$ آنگاه
۴	اگر $A[i] = x$ آنگاه
۵	$counter \leftarrow counter + 1$
۶	پایان اگر
۷	$i \leftarrow i + 1$
۸	به مرحله ۳ برو
۹	پایان اگر
۱۰	$counter$ را برگردان

برنامه:

```

1 int Count(double A[ ],
2           int n, double x)
3 {
4     int counter=0,i;
5     for (i=0; i<n; i++)
6         if (A[i] == x)
7             counter++;
8     return counter;
9 }
```



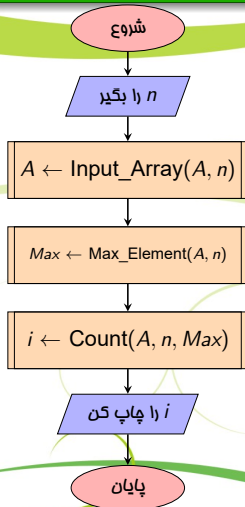
مثال ۱۱

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را محاسبه و چاپ کند.

الگوریتم:

- ۱ شروع
- ۲ n را بگیر
- ۳ $A \leftarrow \text{Input_Array}(A, n)$
- ۴ $Max \leftarrow \text{Max_Element}(A, n)$
- ۵ $i \leftarrow \text{Count}(A, n, Max)$
- ۶ i را چاپ کن
- ۷ پایان





مثال ۱۱

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را محاسبه و چاپ کند.

الگوریتم:

- ۱ شروع
- ۲ n را بگیر
- ۳ $A \leftarrow \text{Input_Array}(A, n)$
- ۴ $\text{Max} \leftarrow \text{Max_Element}(A, n)$
- ۵ $i \leftarrow \text{Count}(A, n, \text{Max})$
- ۶ i را چاپ کن
- ۷ پایان



برنامه:

```
1 int main()  
2 {  
3     unsigned int n,i;  
4     double A[500],Max;  
5     cout << "Enter n: ";  
6     cin >> n;  
7     Input_Array(A,n);  
8     Max=Max_Element(A,n);  
9     i=Count(A,n,Max);  
10    cout<<"# of Max Grades:"  
        <<i;  
11    return 0;  
12 }
```

مثال ۱۱

الگوریتم و فلوچارتی ارائه کنید که ابتدا عدد n و سپس n نمره را بگیرد و تعداد افراد با بیشترین نمره را محاسبه و چاپ کند.

الگوریتم:

- ۱ شروع
- ۲ n را بگیر
- ۳ $A \leftarrow \text{Input_Array}(A, n)$
- ۴ $\text{Max} \leftarrow \text{Max_Element}(A, n)$
- ۵ $i \leftarrow \text{Count}(A, n, \text{Max})$
- ۶ i را چاپ کن
- ۷ پایان



آرایهٔ دو بعدی

آرایهٔ دو بعدی

یک آرایه، از تعدادی متغیر **هم‌نوع** تشکیل شده است که در **فضایی به هم پیوسته** در حافظهٔ اصلی ذخیره می‌شوند و دارای ویژگی‌های زیر است:

$A[0][0]$	$A[0][1]$	$A[0][2]$	$A[0][3]$
$A[1][0]$	$A[1][1]$	$A[1][2]$	$A[1][3]$
$A[2][0]$	$A[2][1]$	$A[2][2]$	$A[2][3]$

هر آرایه دو بعدی دارای یک نام و یک طول و یک عرض است که طول آرایه همان تعداد سطوحی آن و عرض تعداد سطوحی آن آرایه است.

به هر عنصر آرایه با طول n و عرض m می‌توان به وسیلهٔ نام آرایه و دو اندیس، یک اندیس بین صفر و $n - 1$ یک اندیس بین صفر و $m - 1$ دسترسی داشت.



آرایهٔ دو بعدی

آرایهٔ دو بعدی

یک آرایه، از تعدادی متغیر **هم‌نوع** تشکیل شده است که در **فضایی به هم پیوسته** در حافظهٔ اصلی ذخیره می‌شوند و دارای ویژگی‌های زیر است:

- هر آرایه دو بعدی دارای یک نام و یک طول و یک عرض است که طول آرایه همان تعداد سطرهای آن و عرض، تعداد ستون‌های آن آرایه است.
- به هر عنصر آرایه با طول n و عرض m می‌توان به وسیلهٔ نام آرایه و دو اندیس، یک اندیس بین صفر و $n - 1$ یک اندیس بین صفر و $m - 1$ دسترسی داشت.

$A[0][0]$	$A[0][1]$	$A[0][2]$	$A[0][3]$
$A[1][0]$	$A[1][1]$	$A[1][2]$	$A[1][3]$
$A[2][0]$	$A[2][1]$	$A[2][2]$	$A[2][3]$



آرایهٔ دو بعدی

آرایهٔ دو بعدی

یک آرایه، از تعدادی متغیر **هم‌نوع** تشکیل شده است که در **فضایی به هم پیوسته** در حافظهٔ اصلی ذخیره می‌شوند و دارای ویژگی‌های زیر است:

- هر آرایه دو بعدی دارای یک نام و یک طول و یک عرض است که طول آرایه همان تعداد سطرهای آن و عرض، تعداد ستون‌های آن آرایه است.
- به هر عنصر آرایه با طول n و عرض m می‌توان به وسیلهٔ نام آرایه و دو اندیس، یک اندیس بین صفر و $n - 1$ یک اندیس بین صفر و $m - 1$ دسترسی داشت.

$A[0][0]$	$A[0][1]$	$A[0][2]$	$A[0][3]$
$A[1][0]$	$A[1][1]$	$A[1][2]$	$A[1][3]$
$A[2][0]$	$A[2][1]$	$A[2][2]$	$A[2][3]$



مثال ۱۲

زیرالگوریتم گرفتن عناصر یک آرایهٔ دو بعدی با m سطر
و n ستون.



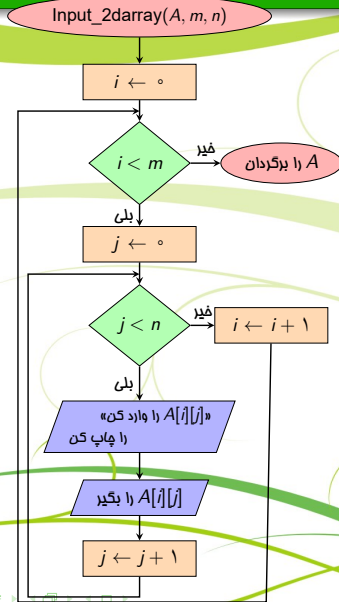
مثال ۱۲

زیرالگوریتم گرفتن عناصر یک آرایه دو بعدی با m سطر و n ستون.

الگوریتم:

۱	شروع $Input_2darray(A, m, n)$
۲	$i \leftarrow 0$
۳	اگر $i < m$ آنگاه
۴	$j \leftarrow 0$
۵	اگر $j < n$ آنگاه
۶	« $A[i][j]$ را وارد کن» را چاپ کن
۷	$A[i][j]$ را بگیر
۸	$j \leftarrow j + 1$
۹	برو به مرحله ۵
۱۰	پایان اگر
۱۱	$i \leftarrow i + 1$
۱۲	به مرحله ۳ برو
۱۳	پایان اگر
۱۴	A را برگردان





مثال ۱۲

زیرالگوریتم گرفتن عناصر یک آرایهٔ دو بعدی با m سطر و n ستون.

الگوریتم:

- ۱ شروع Input_2darray(A, m, n)
- ۲ $i \leftarrow 0$
- ۳ اگر $i < m$ آنگاه
- ۴ $j \leftarrow 0$
- ۵ اگر $j < n$ آنگاه
- ۶ « $A[i][j]$ را وارد کن» را چاپ کن
- ۷ « $A[i][j]$ را بگیر»
- ۸ $j \leftarrow j + 1$
- ۹ برو به مرحلهٔ ۵
- ۱۰ پایان اگر
- ۱۱ $i \leftarrow i + 1$
- ۱۲ برو به مرحلهٔ ۳
- ۱۳ پایان اگر
- ۱۴ A را برگردان



مثال ۱۲

زیرالگوریتم گرفتن عناصر یک آرایه دو بعدی با m سطر و n ستون.

الگوریتم:

۱	شروع $Input_2darray(A, m, n)$
۲	$i \leftarrow 0$
۳	اگر $i < m$ آنگاه
۴	$j \leftarrow 0$
۵	اگر $j < n$ آنگاه
۶	« $A[i][j]$ را وارد کن» را چاپ کن
۷	$A[i][j]$ را بگیر
۸	$j \leftarrow j + 1$
۹	برو به مرحله ۵
۱۰	پایان اگر
۱۱	$i \leftarrow i + 1$
۱۲	به مرحله ۳ برو
۱۳	پایان اگر
۱۴	A را برگردان

برنامه:

```

1 void Input_2darray(double A[100][100],
2                     int m,int n)
3 {
4     int i,j;
5     for(i=0; i<m; i++)
6         for(j=0; j<n; j++)
7             {
8                 cout<<"\nEnter A["<<i<<"]["<<j<<": ";
9                 cin>>A[i][j];
10            }
11 }
```



مثال ۱۳

زیرالگوریتم و زیرفلوچارتی رسم کنید که دو ماتریس $m \times n$ می A و B و ابعاد m و n را به عنوان پارامتر دریافت کند و مجموع A و B یا $A + B$ را محاسبه کرده و برگرداند.



مثال ۱۳

زیرالگوریتم و زیرفلوچارتی رسم کنید که دو ماتریس $m \times n$ A و B و ابعاد m و n را به عنوان پارامتر دریافت کند و مجموع A و B یا $A + B$ را محاسبه کرده و برگرداند.

الگوریتم:

۱	شروع $\text{Matrix_Sum}(A, B, m, n)$
۲	$i \leftarrow 0$
۳	اگر $i < m$ آنگاه
۴	$j \leftarrow 0$
۵	اگر $j < n$ آنگاه
۶	$C[i][j] \leftarrow A[i][j] + B[i][j]$
۷	$j \leftarrow j + 1$
۸	برو به مرحله ۵
۹	پایان اگر
۱۰	$i \leftarrow i + 1$
۱۱	به مرحله ۳ برو
۱۲	پایان اگر
۱۳	C را برگردان



مثال ۱۳

زیرالگوریتم و زیرفلوچارتی رسم کنید که دو ماتریس $m \times n$ سی A و B و ابعاد m و n را به عنوان پارامتر دریافت کند و مجموع A و B یا $A + B$ را محاسبه کرده و برگرداند.

الگوریتم:

۱ شروع $\text{Matrix_Sum}(A, B, m, n)$
۲ $i \leftarrow 0$

۳ اگر $i < m$ آنگاه

۴ $j \leftarrow 0$

۵ اگر $j < n$ آنگاه

۶ $C[i][j] \leftarrow A[i][j] + B[i][j]$

۷ $j \leftarrow j + 1$

۸ برو به مرحله ۵

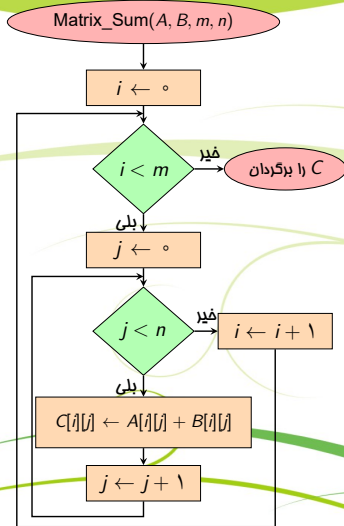
۹ پایان اگر

۱۰ $i \leftarrow i + 1$

۱۱ به مرحله ۳ برو

۱۲ پایان اگر

۱۳ C را برگردان



مثال ۱۳

زیرالگوریتم و زیرفلوچارتی رسم کنید که دو ماتریس $m \times n$ سی A و B و ابعاد m و n را به عنوان پارامتر دریافت کند و مجموع A و B یا $A + B$ را محاسبه کرده و برگرداند.

الگوریتم:

۱	شروع $\text{Matrix_Sum}(A, B, m, n)$
۲	$i \leftarrow 0$
۳	اگر $i < m$ آنگاه
۴	$j \leftarrow 0$
۵	اگر $j < n$ آنگاه
۶	$C[i][j] \leftarrow A[i][j] + B[i][j]$
۷	$j \leftarrow j + 1$
۸	برو به مرحله ۵
۹	پایان اگر
۱۰	$i \leftarrow i + 1$
۱۱	به مرحله ۳ برو
۱۲	پایان اگر
۱۳	C را برگردان

برنامه:

```

1 void Matrix_Sum(
2     double A[100][100],
3     double B[100][100],
4     double C[100][100],
5     unsigned int m,
6     unsigned int n)
7 {
8     unsigned int i, j;
9     for(i=0; i<m; i++)
10         for(j=0; j<n; j++)
11             C[i][j]=A[i][j]+B[i][j];
12 }
```



مثال ۱۴

زیرالگوریتم و زیرفلوچارتی رسم کنید که یک ماتریس $n \times n$ به نام A و n را به عنوان پارامتر دریافت کند و اگر به ازای هر i بین 0 و $n - 1$ ، مجموع عناصر سطر i ام با مجموع عناصر ستون i ام برابر باشد مقدار 1 و در غیر این صورت مقدار 0 را برگرداند.



مثال ۱۴

زیرالگوریتم و زیرفلوچارتی رسم کنید که یک ماتریس $n \times n$ به نام A و n را به عنوان پارامتر دریافت کند و اگر به ازای هر i بین 1 و $n - 1$ ، مجموع عناصر سطر i ام با مجموع عناصر ستون i ام برابر باشد مقدار 1 و در غیر این صورت مقدار 0 را برگرداند.

الگوریتم:

۱	شروع $\text{Sum_Row_Col}(A, n)$
۲	$i \leftarrow 1$
۳	اگر $i < n$ آنگاه
۴	$\text{Sum_Row} \leftarrow 0$; $\text{Sum_Col} \leftarrow 0$; $j \leftarrow 1$
۵	اگر $j < n$ آنگاه
۶	$\text{Sum_Row} \leftarrow \text{Sum_Row} + A[i][j]$
۷	$\text{Sum_Col} \leftarrow \text{Sum_Col} + A[j][i]$
۸	$j \leftarrow j + 1$
۹	به مرحله ۵ برو
۱۰	پایان اگر
۱۱	اگر $\text{Sum_Row} \neq \text{Sum_Col}$ آنگاه
۱۲	0 را برگردان
۱۳	پایان اگر
۱۴	$i \leftarrow i + 1$
۱۵	به مرحله ۳ برو
۱۶	پایان اگر
۱۷	1 را برگردان

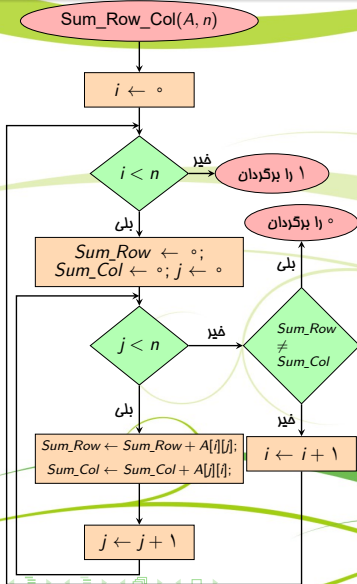


مثال ۱۴

زیرالگوریتم و زیرفلوچارتی رسم کنید که یک ماتریس $n \times n$ به نام A و n را به عنوان پارامتر دریافت کند و اگر به ازای هر i بین 0 و $n-1$ ، مجموع عناصر سطر i ام با مجموع عناصر ستون i ام برابر باشد مقدار 1 و در غیر این صورت مقدار 0 را برگرداند.

الگوریتم:

۱	شروع $Sum_Row_Col(A, n)$
۲	$i \leftarrow 0$
۳	اگر $i < n$ آنگاه
۴	$Sum_Row \leftarrow 0; Sum_Col \leftarrow 0; j \leftarrow 0$
۵	اگر $j < n$ آنگاه
۶	$Sum_Row \leftarrow Sum_Row + A[i][j]$
۷	$Sum_Col \leftarrow Sum_Col + A[j][i]$
۸	$j \leftarrow j + 1$
۹	به مرحله ۵ برو
۱۰	پایان اگر
۱۱	اگر $Sum_Row \neq Sum_Col$ آنگاه
۱۲	0 را برگردان
۱۳	پایان اگر
۱۴	$i \leftarrow i + 1$
۱۵	به مرحله ۳ برو
۱۶	پایان اگر
۱۷	1 را برگردان



مثال ۱۴

زیرالگوریتم و زیرفلوچارتی رسم کنید که یک ماتریس $n \times n$ به نام A و n را به عنوان پارامتر دریافت کند و اگر به ازای هر i بین 0 و $n-1$ ، مجموع عناصر سطر i ام با مجموع عناصر ستون i ام برابر باشد مقدار 1 و در غیر این صورت مقدار 0 را برگرداند.

الگوریتم:

۱	شروع $\text{Sum_Row_Col}(A, n)$
۲	$i \leftarrow 0$
۳	اگر $i < n$ آنگاه
۴	$\text{Sum_Row} \leftarrow 0$; $\text{Sum_Col} \leftarrow 0$; $j \leftarrow 0$
۵	اگر $j < n$ آنگاه
۶	$\text{Sum_Row} \leftarrow \text{Sum_Row} + A[i][j]$
۷	$\text{Sum_Col} \leftarrow \text{Sum_Col} + A[j][i]$
۸	$j \leftarrow j + 1$
۹	به مرحله ۵ برو
۱۰	پایان اگر
۱۱	اگر $\text{Sum_Row} \neq \text{Sum_Col}$ آنگاه
۱۲	0 را برگردان
۱۳	پایان اگر
۱۴	$i \leftarrow i + 1$
۱۵	به مرحله ۳ برو
۱۶	پایان اگر
۱۷	1 را برگردان

برنامه:

```

1 bool Sum_Row_Col(
2     double A[100][100], int n)
3 {
4     int i, j;
5     double Sum_Row=0,
6           Sum_Col=0;
7     for(i=0; i<n; i++)
8     {
9         for(j=0; j<n; j++)
10        {
11            Sum_Row+=A[i][j];
12            Sum_Col+=A[j][i];
13        }
14        if (Sum_Row != Sum_Col)
15            return 0;
16    }
17    return 1;
18 }
```

مثال ۱۵

زیرالگوریتم و زیرفلوچارتی رسم کنید که یک ماتریس $m \times n$ به نام A ، یک ماتریس $n \times k$ به نام B و m و n و k را به عنوان پارامتر بگیرد و $A \times B$ را محاسبه کرده و برگرداند.



مثال ۱۵

زیرالگوریتم و زیرفلوچارتی رسم کنید که یک ماتریس $m \times n$ به نام A ، یک ماتریس $n \times k$ به نام B و m و n و k را به عنوان پارامتر بگیرد و $A \times B$ را محاسبه کرده و برگرداند.

الگوریتم:

شروع	۱
$i \leftarrow 0$	۲
اگر $i < m$ آنگاه	۳
$j \leftarrow 0$	۴
اگر $j < k$ آنگاه	۵
$\ell \leftarrow 0$	۶
اگر $\ell < n$ آنگاه	۷
$C[i][j] \leftarrow C[i][j] + (A[i][\ell] \times B[\ell][j])$	۸
$\ell \leftarrow \ell + 1$	۹
برو به مرحله ۸	۱۰
پایان اگر	۱۱
$j \leftarrow j + 1$	۱۲
برو به مرحله ۵	۱۳
پایان اگر	۱۴
$i \leftarrow i + 1$	۱۵
به مرحله ۳ برو	۱۶
پایان اگر	۱۷
C را برگردان	۱۸



مثال ۱۵

زیرالگوریتم و زیرفلوچارتی رسم کنید که یک ماتریس $m \times n$ به نام A ، یک ماتریس $n \times k$ به نام B و m و n و k را به عنوان پارامتر بگیرد و $A \times B$ را محاسبه کرده و برگرداند.

الگوریتم:

۱ شروع Matrix_Multi(A, B, m, n, k)

۲ $i \leftarrow 0$ ۳ اگر $i < m$ آنگاه۴ $j \leftarrow 0$ ۵ اگر $j < k$ آنگاه۶ $l \leftarrow 0$; $C[i][j] \leftarrow 0$ ۷ اگر $l < n$ آنگاه۸ $C[i][j] \leftarrow C[i][j] + (A[i][l] \times B[l][j])$ ۹ $l \leftarrow l + 1$

۱۰ برو به مرحله ۸

۱۱ پایان اگر

۱۲ $j \leftarrow j + 1$

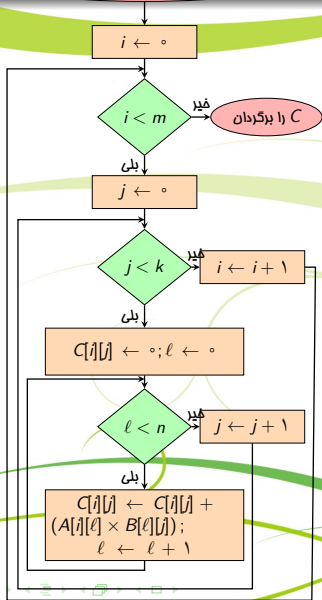
۱۳ برو به مرحله ۵

۱۴ پایان اگر

۱۵ $i \leftarrow i + 1$

۱۶ به مرحله ۳ برو

۱۷ پایان اگر

۱۸ C را برگردان

برنامه:

```

1 void Matrix_Multi(
2     double A[ ][100],
3     double B[ ][100],
4     double C[ ][100],
5     unsigned int m,
6     unsigned int n,
7     unsigned int k)
8 {
9     unsigned int i, j, l;
10    for(i=0; i<m; i++)
11        for(j=0; j<k; j++)
12            {
13                C[i][j]=0;
14                for(l=0; l<n; l++)
15                    C[i][j]+=(A[i][l]*B[l][j]);
16            }
17 }

```

مثال ۱۵

زیرالگوریتم و زیرفلوچارتی رسم کنید که یک ماتریس $m \times n$ به نام A ، یک ماتریس $n \times k$ به نام B و m و n و k را به عنوان پارامتر بگیرد و $A \times B$ را محاسبه کرده و برگرداند.

الگوریتم:

شروع	۱	Matrix_Multi(A, B, m, n, k)
$i \leftarrow 0$	۲	
اگر $i < m$ آنگاه	۳	
$j \leftarrow 0$	۴	
اگر $j < k$ آنگاه	۵	
$\ell \leftarrow 0$; $C[i][j] \leftarrow 0$	۶	
اگر $\ell < n$ آنگاه	۷	
$C[i][j] \leftarrow C[i][j] + (A[i][\ell] \times B[\ell][j])$	۸	
$\ell \leftarrow \ell + 1$	۹	
برو به مرحله ۸	۱۰	
پایان اگر	۱۱	
$j \leftarrow j + 1$	۱۲	
برو به مرحله ۵	۱۳	
پایان اگر	۱۴	
$i \leftarrow i + 1$	۱۵	
به مرحله ۳ برو	۱۶	
پایان اگر	۱۷	
C را برگردان	۱۸	

شیوه ذخیره‌سازی آرایه‌ها در حافظه

شیوه ذخیره‌سازی آرایه‌ها در حافظه

در تعریف آرایه‌ها دو خاصیت برای آنها ذکر شد: اول این که تمام درایه‌ها هم نوع باشند و دوم این که تمام درایه‌های آرایه در یک فضای به هم پیوسته از حافظهٔ کامپیوتر ذخیره شود.

هر سلول حافظه یا هر بایت در حافظهٔ اصلی دارای یک آدرس است و در صورت داشتن آدرس یک محل از حافظه، در زمان ثابتی می‌توان به آن عنصر حافظه دسترسی داشت. به عبارت دیگر، زمان دسترسی به تمام عناصر حافظه اصلی یکسان است و به محل آن بستگی ندارد. این خاصیت را خاصیت دسترسی تصادفی (RAM) حافظه اصلی نامیدیم.

دسترسی به عناصر آرایه نیز تصادفی است و زمان دسترسی به محل قرار گرفتن درایه بستگی ندارد.



شیوه ذخیره‌سازی آرایه‌ها در حافظه

شیوه ذخیره‌سازی آرایه‌ها در حافظه

در تعریف آرایه‌ها دو خاصیت برای آنها ذکر شد: اول این که تمام درایه‌ها هم نوع باشند و دوم این که تمام درایه‌های آرایه در یک فضای به هم پیوسته از حافظه کامپیوتر ذخیره شود.

هر سلول حافظه یا هر بایت در حافظهٔ اصلی دارای یک آدرس است و در صورت داشتن آدرس یک محل از حافظه، در زمان ثابتی می‌توان به آن عنصر حافظه دسترسی داشت. به عبارت دیگر، زمان دسترسی به تمام عناصر حافظه اصلی یکسان است و به محل آن بستگی ندارد. این خاصیت را خاصیت دسترسی تصادفی (RAM) حافظه اصلی نامیدیم.

دسترسی به عناصر آرایه نیز تصادفی است و زمان دسترسی به محل قرار گرفتن درایه بستگی ندارد.



شیوه ذخیره‌سازی آرایه‌ها در حافظه

شیوه ذخیره‌سازی آرایه‌ها در حافظه

در تعریف آرایه‌ها دو خاصیت برای آنها ذکر شد: اول این که تمام درایه‌ها هم نوع باشند و دوم این که تمام درایه‌های آرایه در یک فضای به هم پیوسته از حافظه کامپیوتر ذخیره شود.

هر سلول حافظه یا هر بایت در حافظهٔ اصلی دارای یک آدرس است و در صورت داشتن آدرس یک محل از حافظه، در زمان ثابتی می‌توان به آن عنصر حافظه دسترسی داشت. به عبارت دیگر، زمان دسترسی به تمام عناصر حافظه اصلی یکسان است و به محل آن بستگی ندارد. این خاصیت را خاصیت دسترسی تصادفی (RAM) حافظه اصلی نامیدیم.

دسترسی به عناصر آرایه نیز تصادفی است و زمان دسترسی به محل قرار گرفتن درایه بستگی ندارد.



شیوه ذخیره‌سازی آرایه‌ها در حافظه

شیوه ذخیره‌سازی آرایه‌ها در حافظه

x : آدرس شروع مملى كه آرایه یک بعدی A در حافظه ذخیره شده است

s : تعداد بایت‌های مورد نیاز هر عنصر آرایه

با توجه به ذخیره شدن عناصر آرایه در یک فضای به هم پیوسته، آدرس عنصر $A[i]$ با یک محاسبات سرانگشتی برابر با $x + i \times s$ است.

حافظه اصلی کامپیوتر یک بعدی است، یعنی مشابه یک آرایه یک بعدی است. لذا باید برای ذخیره‌سازی یک ساختار دو بعدی در این فضای یک بعدی ترفندی اندیشیده شود.

روش ذخیره‌سازی سطری، ابتدا سطر اول آرایه دو بعدی ذخیره می‌شود. سپس و بلافاصله بعد از آن سطر دوم آرایه ذخیره می‌شود و الی آخر.

درایه سطر i ام و ستون j ام آرایه دو بعدی در محل $x + (i \times cols + j) \times s$ ذخیره می‌شود، که $cols$ تعداد ستون‌های آرایه دو بعدی است.



شیوه ذخیره‌سازی آرایه‌ها در حافظه

شیوه ذخیره‌سازی آرایه‌ها در حافظه

x : آدرس شروع مملى كه آرایه یک بعدی A در حافظه ذخیره شده است

s : تعداد بایت‌های مورد نیاز هر عنصر آرایه

با توجه به ذخیره شدن عناصر آرایه در یک فضای به هم پیوسته، آدرس عنصر $A[i]$ با یک محاسبات سرانگشتی برابر با $x + i \times s$ است.

حافظه اصلی کامپیوتر یک بعدی است، یعنی مشابه یک آرایه یک بعدی است. لذا باید برای ذخیره‌سازی یک ساختار دو بعدی در این فضای یک بعدی ترفندی اندیشیده شود.

روش ذخیره‌سازی سطری، ابتدا سطر اول آرایه دو بعدی ذخیره می‌شود. سپس و بلافاصله بعد از آن سطر دوم آرایه ذخیره می‌شود و الی آخر.

درایه سطر i ام و ستون j ام آرایه دو بعدی در محل $x + (i \times cols + j) \times s$ ذخیره می‌شود، که $cols$ تعداد ستون‌های آرایه دو بعدی است.



شیوه ذخیره‌سازی آرایه‌ها در حافظه

شیوه ذخیره‌سازی آرایه‌ها در حافظه

x : آدرس شروع مملى كه آرایه یک بعدی A در حافظه ذخیره شده است

s : تعداد بایت‌های مورد نیاز هر عنصر آرایه

با توجه به ذخیره شدن عناصر آرایه در یک فضای به هم پیوسته، آدرس عنصر $A[i]$ با یک محاسبات سرانگشتی برابر با $x + i \times s$ است.

حافظه اصلی کامپیوتر یک بعدی است، یعنی مشابه یک آرایه یک بعدی است. لذا باید برای ذخیره‌سازی یک ساختار دو بعدی در این فضای یک بعدی ترفندی اندیشیده شود.

روش ذخیره‌سازی سطری، ابتدا سطر اول آرایه دو بعدی ذخیره می‌شود. سپس و بلافاصله بعد از آن سطر دوم آرایه ذخیره می‌شود و الی آخر.

درایه سطر i ام و ستون j ام آرایه دو بعدی در محل $x + (i \times cols + j) \times s$ ذخیره می‌شود، که $cols$ تعداد ستون‌های آرایه دو بعدی است.



شیوه ذخیره‌سازی آرایه‌ها در حافظه

شیوه ذخیره‌سازی آرایه‌ها در حافظه

x : آدرس شروع مملى كه آرایه یک بعدی A در حافظه ذخیره شده است

s : تعداد بایت‌های مورد نیاز هر عنصر آرایه

با توجه به ذخیره شدن عناصر آرایه در یک فضای به هم پیوسته، آدرس عنصر $A[i]$ با یک محاسبات سرانگشتی برابر با $x + i \times s$ است.

حافظه اصلی کامپیوتر یک بعدی است، یعنی مشابه یک آرایه یک بعدی است. لذا باید برای ذخیره‌سازی یک ساختار دو بعدی در این فضای یک بعدی ترفندی اندیشیده شود.

روش ذخیره‌سازی سطری، ابتدا سطر اول آرایه دو بعدی ذخیره می‌شود. سپس و بلافاصله بعد از آن سطر دوم آرایه ذخیره می‌شود و الی آخر.

درایه سطر i ام و ستون j ام آرایه دو بعدی در محل $x + (i \times cols + j) \times s$ ذخیره می‌شود، که $cols$ تعداد ستون‌های آرایه دو بعدی است.



شیوه ذخیره‌سازی آرایه‌ها در حافظه

شیوه ذخیره‌سازی آرایه‌ها در حافظه

x : آدرس شروع مملى كه آرایه یک بعدی A در حافظه ذخیره شده است

s : تعداد بایت‌های مورد نیاز هر عنصر آرایه

با توجه به ذخیره شدن عناصر آرایه در یک فضای به هم پیوسته، آدرس عنصر $A[i]$ با یک محاسبات سرانگشتی برابر با $x + i \times s$ است.

حافظه اصلی کامپیوتر یک بعدی است، یعنی مشابه یک آرایه یک بعدی است. لذا باید برای ذخیره‌سازی یک ساختار دو بعدی در این فضای یک بعدی اندیشیده شود.

روش ذخیره‌سازی سطری، ابتدا سطر اول آرایه دو بعدی ذخیره می‌شود. سپس و بلافاصله بعد از آن سطر دوم آرایه ذخیره می‌شود و الی آخر.

درایه سطر i ام و ستون j ام آرایه دو بعدی در محل $s \times (i \times cols + j) + x$ ذخیره می‌شود، که $cols$ تعداد ستون‌های آرایه دو بعدی است.



با آرزوی موفقیت و بهروزی

mfarshi@yazd.ac.ir