



فصل نهم

«آرایه ها»

دکتر فرناز قاسمی
دانشگاه صنعتی امیرکبیر
ترم اول سال تحصیلی ۹۶-۹۵

- ۷) حلقه ها و پرش ها for, While, ...
- ۸) توابع
- ۹) آرایه های یک بعدی و چند بعدی
- ۱۰) رشته ها و نمونه دستیابی به آنها
- ۱۱) کار با فایل ها
- ۱۲) اشاره گر ها
- ۱۳) ساختار ها و یونیون ها



اهتمالی

مقدمه

متغیر: بخشی از حافظه با یک نام که می توان مقداری در آن ذخیره نمود.
(برنامه های کوچک)

آرایه: پردازش های دسته ای

متغیری با یک نام ولی چندین مقدار به طور همزمان از یک نوع
ویرایش محتویات و پردازش داده ها به صورت سریع



مقدمه آرایه (متغیرهای اندیس دار) Array

تعریف: مجموعه ای است از عناصر با طولی ثابت که همگی از یک نوع هستند. (همه خانه های یک آرایه از یک جنس می باشد یعنی پیکانچه آرایه از نوع **int** تعریف شود، ذخیره کردن مقادیر اعشاری موجب از بین رفتن قسمت اعشار فواید شد).

نمونه استفاده:

```
type array_name[array_size];
```

خانه های آرایه به صورت متوالی در حافظه قرار می گیرند.

اندیس اولین خانه آرایه برابر ۰ و اندیس آخرین خانه، ۱ واحد از طول آرایه کمتر است.

طول آرایه در ابتدای برنامه به صورت ثابت تعریف می شود و در طول اجرای برنامه تغییر نمی کند.

آرایه یک سافتمان داده ایست می باشد.

از مهمترین کاربردهای آرایه می توان به استفاده آنها در روشهای مرتب سازی و جستجو اشاره کرد.



عنصر $a[0]$ حاوی مقدار 17.5 و عنصر $a[1]$ حاوی 19.0 و عنصر $a[4]$ حاوی مقدار 18.0 است. این مقادیر می توانند نمرات یک دانشجو در یک نیمسال تحصیلی را نشان دهند.

a

0	17.50
1	19.00
2	16.75
3	15.00
4	18.00

نام آرایه ها: مانند نام متغیرهای دیگر (تنها شامل حرف ها، رقم ها و زیرواژه ها) اندیس (index) یا زیرنویس (subscript) باید یک عدد صحیح یا یک عبارت صحیح باشد. اگر در برنامه ای از یک عبارت برای اندیس استفاده شود آنگاه آن عبارت ارزیابی شده و مقدار حاصل، اندیس مربوطه را معین می کند.

مثال ۱: اگر متغیر a برابر 5 و متغیر b برابر 6 باشد آنگاه مقدار زیر را ارزیابی کنید:

$$c[a + b] += 2;$$

2 واحد به عضو $c[11]$ آرایه اضافه می کند.

$c[0]$	-45
$c[1]$	6
$c[2]$	0
$c[3]$	72
$c[4]$	1543
$c[5]$	-89
$c[6]$	0
$c[7]$	62
$c[8]$	-3
$c[9]$	1
$c[10]$	6453
$c[11]$	78

$$c[0] + c[1] + c[2] = -39$$

تفاوت بین

 n^{th} element of the array: آرایه: n امین عضوو عضو n آرایه: array element n 

۷ پنربری

۶ مرتب سازی

۵ جستجوی
دوگانه

۴ جستجوی فنی

۳ انتقال به تابع

۲ نمونه استفاده

۱ مقدمه

مقداردهی آرایه ها

 $\text{int } x[3] = \{1, -1, 4\}$

۱	-۱	۴
---	----	---

 $\text{float } a[] = \{22.2, 44.4, 66.6\};$

	a
0	22.2
1	44.4
2	66.6

 $\text{int } x[] = \{1, 4, 9, 10\}$

۱	۴	۹	۱۰
---	---	---	----

 $\text{int } x[3] = \{1, 4, 5, -2\}$ $\text{int } x[3] = \{1\}$

۱	.	.
---	---	---

برای تعیین طول آرایه، بهتر است از دستور **define** استفاده کنیم.

```
#define n 100

int main ()
{
    int x[n];
}
```

مزیت: اگر بخواهیم طول آرایه را تغییر دهیم با استفاده از **n** به راحتی امکان پذیر است.

تذکره: طول آرایه را نمی توان به صورت یک متغیر توصیف کرد.

```
main ( )
{
    int x[n], n ; //syntax error
    cin >> n ;
}
```

مثال ۲) برنامه ای بنویسید که از کاربر ۵ عدد را گرفته و به ترتیب معکوس چاپ نماید.

```
int main()
{
    const int SIZE=5; // defines the size N for 5 elements
    double a[SIZE]; // declares the array's elements as type double
    cout << "Enter " << SIZE << " numbers:\t";
    for (int i=0; i<SIZE; i++)
        cin >> a[i];
    cout << "\n In reverse order: ";
    for (int i=SIZE-1; i>=0; i--)
        cout << "\t" << a[i];
    cout << endl;
}
```

Enter 5 numbers: 11.11 33.33 55.55 77.77 99.99
In reverse order: 99.99 77.77 55.55 33.33 11.11

آرایه ها را می توان با استفاده از عملگر جایگزینی مقداردهی کرد اما نمی توان مقدار آنها را به یکدیگر تفصیص داد:

```
float a[7] = { 22.2, 44.4, 66.6 };
float b[7] = { 33.3, 55.5, 77.7 };
b = a; // ERROR: arrays cannot be assigned!
```

همچنین نمی توانیم یک آرایه را به طور مستقیم برای مقداردهی به آرایه دیگر استفاده کنیم:

```
float a[7] = { 22.2, 44.4, 66.6 };
float b[7] = a; // ERROR: arrays cannot be used as initializers!
```

مثال ۳) برنامه ای بنویسید که n را از کار برگرفته و سپس n عدد پرسیده و در یک آرایه ذخیره کند ($n \leq 1000$). در انتها حاصل جمع و بزرگترین عنصر آرایه را چاپ نماید.

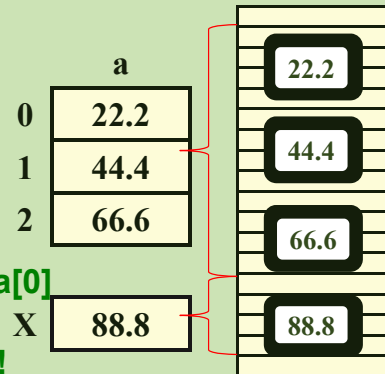
```
#include <iostream>;
#include <conio.h>;
using namespace std;
int main ()
{ int a[1000], n, i, sum = 0, max;
  cout << "Enter n:";
  cin >> n;
  for (i=0 ; i<n ; i++)
  { cout << "Enter number "<< i+1 << " : ";
    cin >> a[i];
  }
  for (i=0 ; i<n ; i++)
    sum += a[i];
  max = a[0];
  for (i=1 ; i<n ; i++)
    if (max < a[i])
      max = a[i];
  cout << "sum of all numbers = " << sum << endl;
  cout << "max of all numbers = " << max << endl;
  getch();
  return 0;
}
```

0	1	2	3	4
9	6	3	-1	5

i	sum	Max	A[i]
0	0	9	9
1	9		6
2	15		3
3	18		-1
4	17		5
5	22		

مثال ۴) خروجی برنامه زیر چیست؟

```
int main()
{
    const int SIZE=4;
    float a[] = { 22.2, 44.4, 66.6 };
    float x=11.1;
    cout << "x = " << x << endl;
    cout << "a = " << a << endl; // the address of a[0]
    cout << "a[0] = " << a[0]; // the value of a[0]
    a[3] = 88.8; // ERROR: index is out of bounds!
    cout << "x = " << x << endl;
}
```



X=11.1
a = 0x0064fdec
a[0] = 22.2
x = 11.1 ? 88.8

```
#include <iostream>;
using namespace std;
int main ()
{
    int a[1000], i, max1, max2, n;
    cout<<"Enter n: ";
    cin>>n;
    for (i=0 ; i<n ; i++)
    {
        cout<<"Enter a#" <<i+1<<":";
        cin>>a[i] ;
    }
    max1=a[0] ;
    max2=a[1] ;
    for (i=1 ; i<n ; i++)
    {
        if (a[i]>max1)
        {
            max2= max1 ;
            max1= a[i] ;
        }
        else if (a[i]>max2)
            max2=a[i];
    }
    cout<<max1<<","<<max2;
    return 0 ;
}
```

مثال ۵) برنامه ای بنویسید که n را گرفته و از کاربر n عدد بخواهد و دو عدد بزرگتر از بقیه اعداد را بیابد (n≤1000)

۷ پندبیری

۶ مرتب سازی

۵ جستجوی دودویی

۴ جستجوی قطعی

۳ انتقال به تابع

۲ نمونه استفاده

۱ مقرر

مثال ۶) برنامه ای بنویسید که آرایه ای بطول ۲۰، توسط کاربر مقداردهی کند و سپس ترتیب عناصر آرایه را جابه جا کند یعنی عنصر اندیس صفر را با عنصر اندیس ۱۹ و عنصر اندیس ۱ را با عنصر اندیس ۱۸ جابه جا کند؟

#include <iostream>;

using namespace std;

int main ()

{

int a[20], i;

for (i = 0; i < 20; i++)

{

cout << "enter number " << i + 1 << ": ";

cin >> a [19-i];

}

for (i = 0; i < 20 ; i++)

cout << i << " , " << a[i] << endl;

}

۳	۹	۱۲	۱۵
---	---	----	----

۱۵	۱۲	۹	۳
----	----	---	---

15/39

دانشگاه صنعتی امیرکبیر - دکتر قاسمی

12/20/2016

۷ پندبیری

۶ مرتب سازی

۵ جستجوی دودویی

۴ جستجوی قطعی

۳ انتقال به تابع

۲ نمونه استفاده

۱ مقرر

#include <iostream>

#include <conio.h>

#define n 50

using namespace std;

int main()

{

int A[n], pos[n], neg[n];

int i, k=0, l=0;

for(i=0; i<n; i++)

{

cout << "A[" << i << "]=";

cin >> A[i];

}

for(i=0; i<n; i++)

if (A[i]>=0)

pos[k++] = A[i];

else

neg[l++] = A[i];

for (i=0; i<k; i++)

cout << "\npos[" << i << "]=" << pos[i];

cout << "\n";

for(i=0; i<l; i++)

cout << "\nneg[" << i << "]=" << neg[i];

getch();

return(0);

}

مثال ۷) آرایه ای از اعداد صحیح به نام A و طول n = 50 در نظر بگیرید.

برنامه ای بنویسید که اعضای این آرایه را از ورودی دریافت و اعداد منفی و مثبت آرایه را از هم جدا کرده و اعداد مثبت را در دافل آرایه pos و اعداد منفی را در دافل آرایه neg قرار دهد.

16/39

دانشگاه صنعتی امیرکبیر - دکتر قاسمی

12/20/2016

۷. پندبیری

۶. مرتب سازی

۵. جستجوی دودویی

۴. جستجوی فطی

۳. انتقال به تابع

۲. نمونه استفاده

۱. مقرر مه

```
#include<iostream>
#include<conio.h>
#define size 5
using namespace std;
int main()
{
    int A[size];
    int i,j;
    for(i=0;i<size;i++)
    {
        cout<<" Enter A["<<i<<"]="";
        cin>>A[i];
    }
    cout<<"Elements";
    for(i=0;i<size;i++)
    {
        cout<<"\n"<<i;
        for(j=1;j<=A[i];j++)
            cout<<"*";
    }
    cout<<"\n";
    getch();
    return(0);
}
```

مثال ۱) برنامه ای بنویسید که پنج عدد صحیح از ورودی دریافت و معادل عدد دریافتی یک نمودار میله ای نماید، یعنی متناسب با هر یک از اعداد خوانده شده، میله ای از ستاره ها به طول عدد ورودی چاپ نماید. خروجی این برنامه برای داده های ورودی به ترتیب ۱۲، ۳، ۱، ۶ و ۱۵ به شکل باشد:

```
0 * * * * *
1 * * *
2 * * * * *
3 * * * * *
4 * * * * *

```

17/12/20/2016

دانشگاه صنعتی امیرکبیر - دکتر قاسمی

12/20/2016

۷. پندبیری

۶. مرتب سازی

۵. جستجوی دودویی

۴. جستجوی فطی

۳. انتقال به تابع

۲. نمونه استفاده

۱. مقرر مه

```
#include <iostream>
#include <stdlib.h>
#include <time.h>
#define SIZE 7
using namespace std;
int main ( void )
{
    int face;
    int roll;
    int frequency [SIZE ] = { 0 };
    srand(time(NULL));
    for ( roll = 1; roll <= 6000; roll++ )
    {
        face = 1 + rand( ) % 6;
        ++ frequency[ face ];
    }
    cout << "Face" << "\t" << " frequency " << endl;
    for ( face = 1; face < SIZE; face++ )
        cout << face<< "\t" << frequency[face]<< endl;
    return 0;
}
```

مثال ۹) پرتاب یک تاس به تعداد 6000 مرتبه و ارائه خلاصه نتایج در داخل یک آرایه

18/12/20/2016

دانشگاه صنعتی امیرکبیر - دکتر قاسمی

12/20/2016

فروچی:

Face	frequency
1	1029
2	951
3	987
4	1033
5	1010
6	990

```
#include <stdio.h>
#include <iostream>
#define RESPONSESIZE 40 /* define array sizes */
#define FRQUENCYSIZE 11
using namespace std;
int main ( void )
{
    int answer; /* counter to loop through 40 responses */
    int rating; /* counter to loop through frequencies 1-10 */
    int frequency [FRQUENCYSIZE] = { 0 };
    int responses [RESPONSESIZE] = { 1, 2, 6, 4, 8, 5, 9, 7, 8, 10, 1,
    6, 3, 8, 6, 10, 3, 8, 2, 7, 6, 5, 7, 6, 8, 6, 7, 5, 6, 5, 6, 7, 5, 6, 4, 8, 6,
    8, 10 };
    for (answer = 0; answer < RESPONSESIZE; answer++)
        ++ frequency[responses[answer]];
    cout<< "Rating" << "\t" << "Frequency"<< endl;
    for ( rating = 1; rating < FRQUENCYSIZE; rating++ )
        { cout<< rating << "\t" << frequency[rating] << endl; }
    return 0;
}
```

مثال ۴۰: از دانشگاه

پرسیده می شود تا نظر خود را در مورد کیفیت غذای سالن غذاخوری

اساس نمره ۱ تا ۱۰ (۱ به

معنی خیلی بد و ۱۰ به معنی

خیلی خوب) بیان کنند. ۴۰

جواب دریافت شده را در

یک آرایه صحیح قرار داده و

نتایج حاصل از رأی گیری را

ارائه دهید.

فروچی:

Rating frequency

1	2
2	2
3	2
4	2
5	5
6	11
7	5
8	7
9	1
10	3

انتقال آرایه ها به توابع



انواع دیگر آرایه ها

آرایه های char (هاوی رشته ها)

پایان دهنده ویژه

انتقال آرایه به تابع: انتقال طول آرایه +

از طریق ارجاع (by reference)

نام آرایه، آدرس اولین عضو آرایه است.

const برای جلوگیری از تغییر مقادیر آرایه در یک تابع

عضوهای آرایه در بدنه تابع، ثابت و بدون تغییر می شوند، و تلاش برای تغییر

یک عضو آرایه در بدنه تابع، موجب خطای زمان کامپایل می شود.

مثال ۱۱) توابع ورودی و خروجی برای یک آرایه

```
#include <iostream>
using namespace std;
void read(int[],int&);
void print(int[],int);
const int MAXSIZE=100;
int main()
{ int a[MAXSIZE]={0}, size;
  read(a,size);
  cout << "The array has " << size << " elements: ";
  print(a,size);
}
void read(int a[], int& n)
{ cout << "Enter integers. Terminate with 0:\n";
  n = 0;
  do
  { cout << "a[" << n << "]: ";
    cin >> a[n];
  } while (a[n++] !=0 && n < MAXSIZE);
  --n; // don't count the 0
}
void print(int a[], int n)
{ for (int i=0; i<n; i++)
  cout << a[i] << " ";
}
```

23/39

دانشگاه صنعتی امیرکبیر - دکتر قاسمی

12/20/2016

کاربردهای آرایه: روشهای جستجو

۱. جستجوی
فطی

۲. جستجوی
دودویی

۳. مرتب سازی
انتقابی و حبابی

جستجوی ترتیبی یا فطی (sequential search)

هدف: جستجوی عنصر **item** در یک آرایه یک بعدی

مقایسه **item** با **تک تک عناصر آرایه**: چنانچه در آرایه یافت شود جریان جستجو با ارسال پیام مثبت قطع می‌شود.

بهترین حالت زمانی است که عنصر مورد جستجو اولین عنصر آرایه برابر باشد جریان جستجو با ارسال پیام مثبت قطع می‌شود.

بدترین حالت زمانی است که عنصر مورد جستجو آخرین عنصر لیست موجود باشد.

این روش جستجو برای لیست‌های نامرتب استفاده می‌شود.

مثال ۱۲) برنامه‌ای بنویسید که آرایه‌ای بطول ۱۰ را به صورت اولیه مقدار دهی کند و با دریافت یک عدد از کاربر تعداد تکرارهای آن عدد را در آن آرایه تعیین و چاپ کند؟

```
#include <iostream>
#define n 10
using namespace std;
int main ( )
{int A[n], i, found = 0, B = 0;
int item;
for (i = 0; i < n; i++)
    cin >> A[i];
cout << "enter a number to search: ";
cin >> item;
for ( i = 0; i < n; i++)
{    if (item == A[i])
        {found = 1;
        B += 1;
        }
    }
```

۹	۵	۹	۹	۱	۷	۳	۲	۹	۱
---	---	---	---	---	---	---	---	---	---

```
if (found == 1)
    cout << " Number of items in the list: " << B << endl;
else
    cout << " Item dose not exist in the list"<< endl;
return 0;
}
```

۲. الگوریتم جستجوی دودویی یا باینری

لیست های مرتب:

تقسیم آرایه از وسط به دو بخش بالایی و پایینی

مقایسه مقدار مورد جستجو با اولین عنصر بخش پایینی:

(بر اساس صعودی یا نزولی بودن لیست) کوپکت: مورد جستجو در بخش پایینی

وجود ندارد و باید در بخش بالایی به دنبال آن گشت.

دوباره بخش بالایی به دو بخش تقسیم می گردد و گام فوق تکرار می شود.

در نهایت محدوده جستجو: یک عنصر

مقایسه روش های جستجوی قطعی و دودویی

۱. جستجوی باینری فقط در مورد لیست های مرتب انجام پذیر می باشد و اگر آرایه ای مرتب نباشد، جستجوی دودویی پاسخ غلط می دهد ولی جستجوی قطعی همیشه پاسخ صحیح خواهد داد.

۲. پیچیده تر از روش جستجوی قطعی اما بسیار سریع تر به جواب می رسیم.

در این روش هر بار که حلقه تکرار می شود، محدوده جستجو ۵۰٪ کوپکت می شود. در آرایه n عنصری، روش جستجوی دودویی حداکثر به $\log 2n+1$ مقایسه نیاز دارد تا به پاسخ برسد. حال آنکه در روش جستجوی قطعی به n مقایسه نیاز است.

۳. اگر چند عنصر دارای مقادیر یکسانی باشند، آنگاه جستجوی قطعی همیشه کوپکتترین ایندکس را برمی گرداند ولی در مورد جستجوی دودویی نمی توان گفت که کدام ایندکس بازگردانده می شود.

۷. چندبرری

۶. مرتب سازی

۵. جستجوی دودویی

۴. جستجوی قطعی

۳. انتقال به تابع

۲. نحوه استفاده

۱. مقدمه

include <iostream>

define n 7

using namespace std;

int main ()

{ int beg = 0 , end = n-1 , mid;

int x[n], i, item;

cout << "Enter the list(7 elements): ";

for (i = 0; i < n; i++)

cin >> x[i];

cout << "Enter item to search :";

cin >> item ;

mid = (beg + end) / 2 ;

while (x[mid] != item && beg <= end)

{if (item > x[mid])

beg = mid + 1 ;

else

end = mid - 1 ;

mid = (beg + end) / 2;}

if (beg > end)

cout << " Item does not exist in the list. \n";

else

cout << "Item exists in list with index " << mid << endl;

}

29/39

دانشگاه صنعتی امیرکبیر - دکتر قاسمی

12/20/2016

مثال ۱۳) برنامه ای بنویسید که عددی از کاربر گرفته و در یک آرایه مقداردهی شده به روش دودویی آن را جستجو کند و اندیس آن را چاپ نماید.

۰	۱	۲	۳	۴	۵	۶
۹	۱۲	۱۶	۲۵	۳۶	۴۰	۶۵

۰	۱	۲	۳	۴
۹	۳۷			

	beg	end	mid
۰	۰	۶	۳
۰	۰	۲	۱
۰	۰	۰	۰
۰	۰	۶	۳
۴	۴	۶	۵
	۴	۴	۴
۵	۵	۴	۴

روش های مرتب سازی

اساس آن، مقایسه و جابجایی می باشد.

روش مرتب سازی انتخابی (selection sort):

در این الگوریتم در هر مرحله بزرگترین عنصر لیست پیدا شده و با یک تعویض در محل نهایی خود قرار می گیرد با این روند لیست به صورت صعودی (نزولی) مرتب خواهد شد.

3	-1	-2	2
---	----	----	---

2 -1 -2 3

مثال ۱۴) قطعه برنامه مرتب سازی انتخابی:

```
#include <iostream>
using namespace std;
int main()
{
    int a[] = { 99, 33, 44, 55, 66, 77, 88 };
    int n=7,min, pos,j,i,k ;
    for (i = n-1; i > 0 ; i--)
    { cout<<"a["<<i<<"]": "<<a[i]<<endl;
      min = a[i];
      pos = i ;
      for (j=0 ; j <= i-1 ;j++)
          if (a[j]< min)
              {min = a[j] ;
               pos = j ;}
      cout<<"j: "<<j-1<< "\t";
      for (k=0 ; k < 7 ;k++)
          cout << a[k] << "\t";
      cout<<"\n";
      a[pos] = a[i] ;
      a[i] = min ;
    }
    for (j=0 ; j < n ;j++)
        cout << a[j] << "\t";
    cout<<"\n";
}
```

مرتب سازی ببابی

واژه نامه: مرتب شده بر اساس حروف الفبا
جستجو بسیار سریع تر به شرط آن که همه عناصر **مرتب** باشند.
آرایه پندین مرتبه پوش می شود و در هر مرتبه بزرگترین عنصر موجود به سمت بالا هدایت می شود و سپس محدوده مرتب سازی برای مرتبه بعدی یکی کاسته می شود. در پایان همه پوش ها، آرایه مرتب شده است.

مقایسه و بابایی زوج های همسایه:

مقایسه عنصر اول با عنصر دوم بابایی عنصر بزرگتر به پایین سپس
مقایسه عنصر دوم با عنصر سوم بابایی عنصر بزرگتر به پایین و به همین ترتیب تا انتهای آرایه



بزرگترین عضو آرایه در فانه انتهای
یک واحد کاهش محدوده جستجو و مقایسه زوج های همسایه (انتقال عدد بزرگتر بعدی به انتهای محدوده)
ادامه پوش تا محدود شدن جستجو به عنصر اول (آرایه مرتب شده)

مثال ۱۵ روش bubble

```
#include <iostream>
using namespace std;
int main()
{
    int n, i, j, t, a[1000];
    cout<<"Enter n: ";
    cin>>n;
    for (j=0; j < n; j++)
        {cout<<"Element no:"<<j<<": ";
        cin>>a[j];}
    for (i=n-1; i>0; i--)
        {for (j=0; j<i; j++)
            {if (a[j]>a[j+1])
                {t=a[j];
                 a[j]=a[j+1];
                 a[j+1]=t;}
            }
        }
    for (j=0; j < n; j++)
        cout<<a[j]<<"\t";
}
```

$$n(n-1)/2$$

مرتبه مراحل n^2 است.

25	14	73	88	53	110	40	7
14	25	73	53	88	40	7	110
14	25	53	73	40	7	88	
14	25	53	40	7	73		

آرایه های چند بعدی

آرایه ای از نوع آرایه: یعنی هر خانه از آن آرایه، خود یک آرایه باشد.
دستور: `int a[2][5];` آرایه ای با دو عنصر که هر کدام، یک آرایه پنج عنصری از نوع `int` است.
آرایه سه بعدی: هر خانه از آن یک آرایه دوبعدی است.

<code>a[0][0]</code>	<code>a[0][1]</code>	<code>a[0][2]</code>	<code>a[0][3]</code>	<code>a[0][4]</code>
<code>a[1][0]</code>	<code>a[1][1]</code>	<code>a[1][2]</code>	<code>a[1][3]</code>	<code>a[1][4]</code>

تعداد ستون = y , تعداد سطر = x : `a[x][y]`

دستور: `int a[2][3][5];` آرایه ای با دو عنصر تعریف می کند که هر عنصر، سه آرایه است که هر آرایه پنج عضو از نوع `int` دارد.

قرار دادن مقدار ۹۹ در عنصری با ایندکس (۳و۲و۱)
`a[1][2][3] = 99;`

ارسال آرایه های چندبعدی به توابع:

شبیه ارسال آرایه های یک بعدی
تعیین بعد اول اختیاری است ولی تمام بعدهای بعدی باید مشخص باشند.

```
int test(int A[ ][...])
```

```
{
...
}
```

اشتباه

```
int test(int A[ ][4]...)
```

```
{
...
}
```

درست

۷

پنربعدی

۶

مرتب سازی

۵

جستجوی
دوگانه

۴

جستجوی فطی

۳

انتقال به تابع

۲

نمونه استفاده

۱

مقدمه

مثال ۱۶) نوشتن و خواندن یک آرایه دوبعدی

```
int main()
{
    int a[3][5];
    cout << "Enter 15 integers, 5 per row:\n";
    for (int i=0; i<3; i++)
    {
        cout << "ROW " << i << ": ";
        for (int j=0; j<5; j++)
            cin >> a[i][j];
    }
    for (int i=0; i<3; i++)
    {
        for (int j=0; j<5; j++)
            cout << " " << a[i][j];
        cout << endl;
    }
}
```

Enter 15 integers, 5 per row:

row 0: 44 77 33 11 44

row 1: 60 50 30 90 70

row 2: 65 25 45 45 55

44 77 33 11 44

60 50 30 90 70

65 25 45 45 55

37/39

دانشگاه صنعتی امیرکبیر - دکتر قاسمی

12/20/2016

۷

پنربعدی

۶

مرتب سازی

۵

جستجوی
دوگانه

۴

جستجوی فطی

۳

انتقال به تابع

۲

نمونه استفاده

۱

مقدمه

مثال ۱۷) برنامه ای بنویسید که دو ماتریس $m \times n$ و $n \times k$ را ضرب کرده و ماتریس $m \times k$ را به کاربر نمایش دهد.

```
#include <iostream>
using namespace std;
int main ( )
{
    int a[50][50], b[50][50], c[50][50];
    int m, n, k, i, j, p;
    cout << "Enter m, n, k for matrices: ";
    cin >> m >> n >> k;
    cout << "Enter A(" << m << "x" << n << "): ";
    for ( i=0 ; i<m ; i++)
        for ( j=0 ; j<n ; j++)
            cin >> a[i][j];
    cout << "Enter B(" << n << "x" << k << "): ";
    for(i=0 ; i<n ; i++)
        for(j=0 ; j<k ; j++)
            cin >> b[i][j];
    cout << "A*B=" << endl;

    for (i=0 ; i<m ; i++)
    {
        for (j=0 ; j<k ; j++)
        {
            c[i][j]=0;
            for (p=0 ; p<n ; p++)
                c[i][j]=c[i][j] + a[i][p] * b[p][j];
            cout << c[i][j] << "t";
        }
        cout << endl;
    }
}
```

38/39

دانشگاه صنعتی امیرکبیر - دکتر قاسمی

12/20/2016

پرسش و پاسخ ؟

