

آرایه‌ها ۲

مبانی برنامه‌نویسی

(۱۳۹۱-۱۳۹۰-۱۳۸۹)

جلسه‌ی بیست و نهم



دانشگاه شهید بهشتی

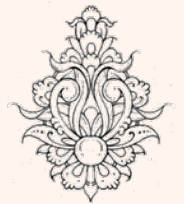
پاییز ۱۳۹۱

دانشکده‌ی مهندسی برق و کامپیوتر

احمد محمودی ازناوه

## فهرست مطالب

- نکاتی در مورد آرایه‌ها
- آرایه‌های موازی
- ارسال آرایه به تابع

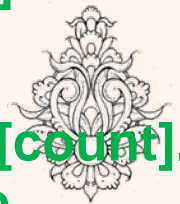


- به روی هر عنصر از آرایه همانند یک متغیر معمولی عملیات ریاضی می تواند صورت پذیرد.

**Pay = hours [3] \* rate ;**

```
int score[5] = { 7 , 8 , 9 , 10 , 11 } ;
++score[2] ; // Pre-increment operation on the value in score [2]
Score[4]++ ; // Post-increment operation on the value in score [4]
```

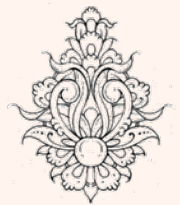
```
amount [count]-- ; //This decrements the value stored in amount [count].
amount [count--] ; //This decrements the variable count, but dose
//nothing to the value stored in amount [count]
```



- گروه‌های دور اندیس آرایه یک عملگر در ++c محسوب می‌گردند.
- تقدم این گروه‌ها با پرانتز برابر است.
- اگر a برابر با ۵ و b برابر با ۶ باشد مقدار عبارت زیر چند است؟

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[ a + b ] += 2;`  
`X=c[6]/2+c[a+b];`



```

int main()
{
const int NUM_WORKERS = 5;
int worker;
int hours[NUM_WORKERS];
double payRate, grossPay;

cout << "Enter the hours worked by " << NUM_WORKERS
<< " employees who all\n"
<< "earn the same hourly rate.\n";

for (worker = 0; worker < NUM_WORKERS; worker++)
{
    cout << "Employee #" << (worker+1) << ": ";
    cin >> hours[worker];
}

cout << "\nEnter the hourly pay rate for all the employees: ";
cin >> payRate;

cout << "\nHere is the gross pay for each employee:\n";

for (worker = 0; worker < NUM_WORKERS; worker++)
{
    double grossPay = hours[worker] * payRate;
    cout << "Employee #" << (worker+1);
    cout << ": $" << grossPay << endl;
}

return 0;
}

```

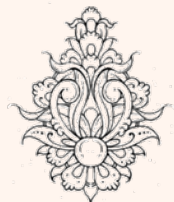
```

Enter the hours worked by 5 employees who all
earn the same hourly rate.
Employee #1: 12
Employee #2: 3
Employee #3: 44
Employee #4: 5
Employee #5: 6

Enter the hourly pay rate for all the employees: 12

Here is the gross pay for each employee:
Employee #1: $144
Employee #2: $36
Employee #3: $528
Employee #4: $60
Employee #5: $72

```

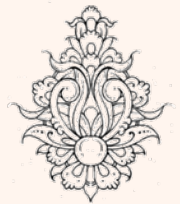


```
#include <iostream>
using namespace std;

int a[] = {1, 2, 3, 4, 40};
int n, result=0;

int main ()
{
    for ( n=0 ; n<5 ; n++ )
    {
        result += a[n];
    }
    cout << "The sum is:" <<result;
    return 0;
}
```

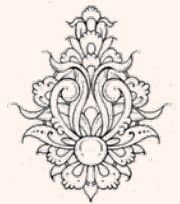
The sum is:50



- اعدادی از یک آرایه را خوانده اطلاعات را به صورت هیستوگرام فراوانی رسم کنید.  
– برای نشان دادن هر عنصر از "\*" استفاده نمایید.

{12,3,4,6,13,16,2,7,0,9}

Element	Value	Histogram
0	12	*****
1	3	***
2	4	****
3	6	*****
4	13	*****
5	16	*****
6	2	**
7	7	*****
8	0	
9	9	*****




# هیستوگرام فراوانی

```
#include <iostream>
using namespace std;
int main()
{
const int arraySize = 10;
int n[ arraySize ]={12,3,4,6,13,16,2,7,0,9}; // array s has 10 elements

cout << "Element\t" << "Value\t"<< "Histogram" << endl;

for ( int i= 0; i < arraySize; i++ )
{
    cout << i << "\t" << n[ i ]<<"\t";
        for ( int j = 0; j < n[i]; j++ )
            cout<< '*';
        cout << endl;
    }
    return 0; // indicates successful termination
} // end main
```



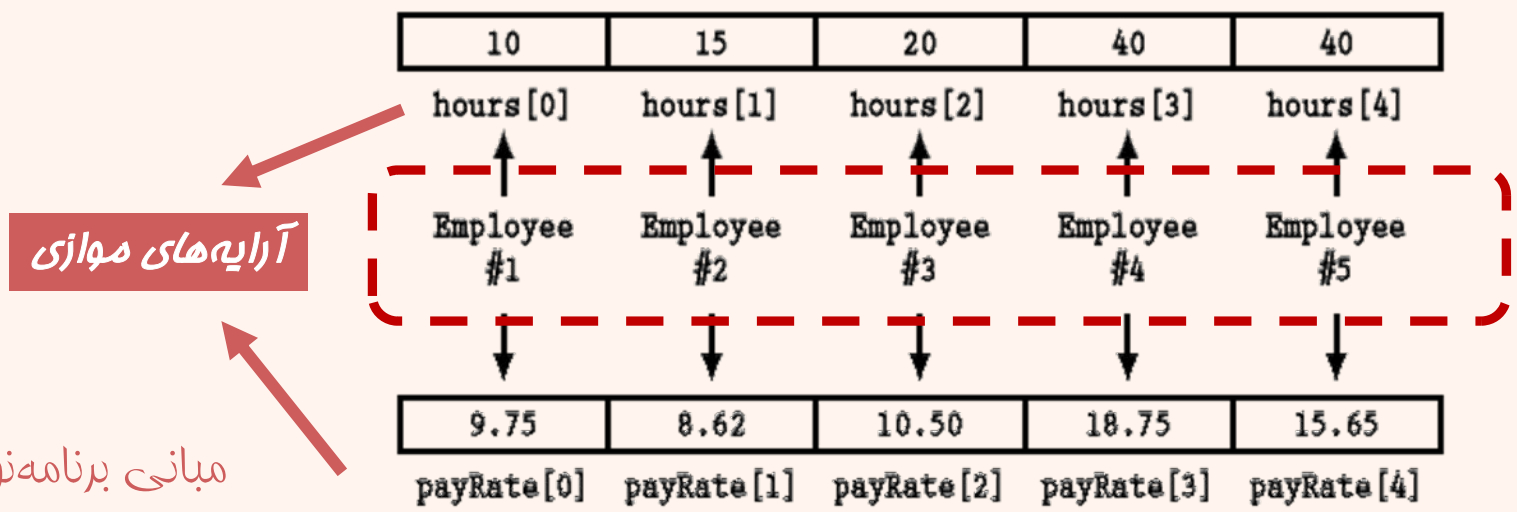
Element	Value	Histogram
0	12	*****
1	3	***
2	4	****
3	6	*****
4	13	*****
5	16	*****
6	2	**
7	7	*****
8	0	
9	9	*****



# استفاده از آرایه‌ی موازی

- در بسیاری موارد ذخیره‌ی داده‌های مرتبط در دو یا چند آرایه می‌تواند بسیار مفید باشد.
- در این صورت به واسطه‌ی استفاده از اندیس یکسان می‌توان به داده‌های مرتبط دست‌یافت.
- مثال

– برنامه‌ی حقوق کارکنان را به گونه‌ای بنویسید که برای هر پرسنل حقوق ساعتی متفاوتی در نظر گرفته شود:



```

int main()
{
    const int NUM_EMPS = 5;
    int index;
    int hours[NUM_EMPS]; // Define 2 parallel arrays
    double payRate[NUM_EMPS];
    double grossPay;
    // Get employee work data
    cout << "Enter the hours worked and hourly pay rates of "
    << NUM_EMPS << " employees. \n";
    for (index = 0; index < NUM_EMPS; index++)
    {
        cout << "Hours worked by employee #" << (index + 1) << ": ";
        cin >> hours[index];
        cout << "Hourly pay rate for employee #" << (index + 1) << ": ";
        cin >> payRate[index];
    }
    // Display the data
    cout << "\nHere is the gross pay for each employee:\n";
    for (index = 0; index < NUM_EMPS; index++)
    {
        grossPay = hours[index] * payRate[index];
        cout << "Employee #" << (index + 1);
        cout << ": $" << grossPay << endl;
    }
    return 0;
}

```

```

Enter the hours worked and hourly pay rates of 5 employees.
Hours worked by employee #1: 4
Hourly pay rate for employee #1: 15
Hours worked by employee #2: 2
Hourly pay rate for employee #2: 56
Hours worked by employee #3: 33
Hourly pay rate for employee #3: 6
Hours worked by employee #4: 23
Hourly pay rate for employee #4: 22
Hours worked by employee #5: 3
Hourly pay rate for employee #5: 44

Here is the gross pay for each employee:
Employee #1: $60
Employee #2: $112
Employee #3: $198
Employee #4: $506
Employee #5: $132

```

مقداردهی دو آرایه



- در بعضی موارد نیاز است آرایه‌ای را به تابعی ارسال نماییم.
- اگر در هر مرحله ارسال تنها یک متغیر از آرایه نیاز باشد همانند ارسال یک متغیر عمل می‌شود.

```

void showValue(int); // Function prototype

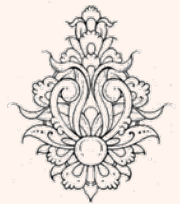
int main()
{
    const int ARRAY_SIZE = 8;
    int collection[ARRAY_SIZE] = {5, 10, 15, 20, 25, 30, 35, 40};

    for (int cycle = 0; cycle < ARRAY_SIZE; cycle++)
        showValue(collection[cycle]);
    cout << endl;
    return 0;
}

void showValue(int num)
{
    cout << num << " ";
}

```

5 10 15 20 25 30 35 40



## ارسال آرایه به تابع

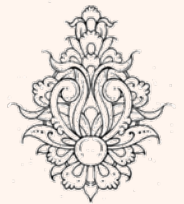
- برای اینکه یک آرایه را به عنوان آرگومان ورودی به تابعی ارسال نماییم لازم است کامپایلر نوع عناصر آرایه و نام آرایه را بداند.
- آن چیزی که در نظر گرفته می‌شود آدرس شروع آرایه و نوع عناصر است.

```
int myarray [40];
```

```
void procedure(int arg[])
```

```
procedure(myarray)
```

← فرآیندی تابع



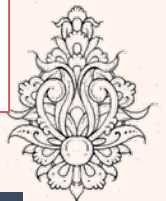
```
void showValues(int [], int); // Function prototype
```

```
int main()  
{  
    const int ARRAY_SIZE = 8;  
    int collection[ARRAY_SIZE] = {5, 10, 15, 20, 25, 30, 35, 40};  
    showValues(collection, ARRAY_SIZE);  
    return 0;  
}
```

```
void showValues (int nums[], int size)
```

```
{  
    for (int index = 0; index < size; index++)  
        cout << nums[index] << " ";  
    cout << endl;  
}
```

5 10 15 20 25 30 35 40



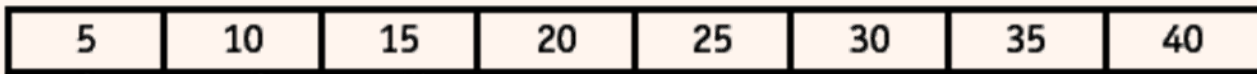
دانشگاه  
سپهر  
بهشتی

ارسال نام آرایه در مقیقت ارسال نشانی اولین عنصر آرایه است  
پس ارسال آرایه به تابع شبیه ارسال به طریق ارجاع عمل می‌کند

```
showValues (collection, ARRAY_SIZE) ;
```

```
void showValues (int nums [ ] , int size)  
{  
    for (int index = 0; index < size ; index++)  
        cout << nums [ index ] << " " ;  
        cout << endl;  
}
```

collection array of eight integers



nums [0]  
references

collection[0]

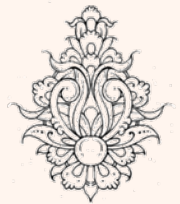
nums [1]  
references

collection[1]

nums [2]  
references

collection[2]

... and so forth



```

#include <iostream>
using namespace std;

void printarray (int arg[], int length) {
    for (int n=0; n<length; n++)
        cout << arg[n] << " ";
    cout << "\n";
}

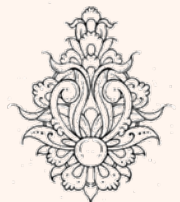
int main ()
{
    int firstarray[] = {5, 10, 15};
    int secondarray[] = {2, 4, 6, 8, 10};
    printarray (firstarray,3);
    printarray (secondarray,5);
    return 0;
}

```

```

5 10 15
2 4 6 8 10

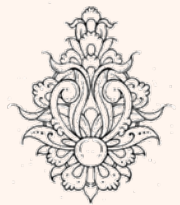
```



```
#include <iostream>
using namespace std;
int sum(int[],int);
int main()
{   int a[] = { 11, 33, 55, 77 };
    int size = sizeof(a)/sizeof(int);
    cout << "sum(a,size) = " << sum(a,size) << endl;
}

int sum(int a[], int n)
{   int sum=0;
    for (int i=0; i<n; i++)
        sum += a[i];
    return sum;
}
```

```
sum(a, size) = 176
```





```

#include <iostream>
using namespace std;
int main()
{
    int a[4] = { 44, 32, 345, 134};
    cout << "a = " << a << endl; // the address of a[0]
    cout << "a[0] = " << a[0]<<endl; // the value of a[0]
}

```

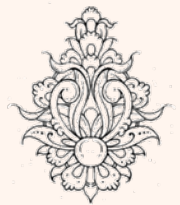
```

a = 0012FF1C
a[0] = 44

```

a

0012FF1C



```

#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 print(int a[],int n)
{
    for (int i=0; i<n; i++)
        cout << a[i] << " ";
}

```

```

Enter integers. Terminate with 0:
a[0]: 12
a[1]: 3
a[2]: 4
a[3]: 5
a[4]: 6
a[5]: 0
The array has 5 elements: 12 3 4 5 6

```

```

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
}

```

مبانی برنامه نویسی

