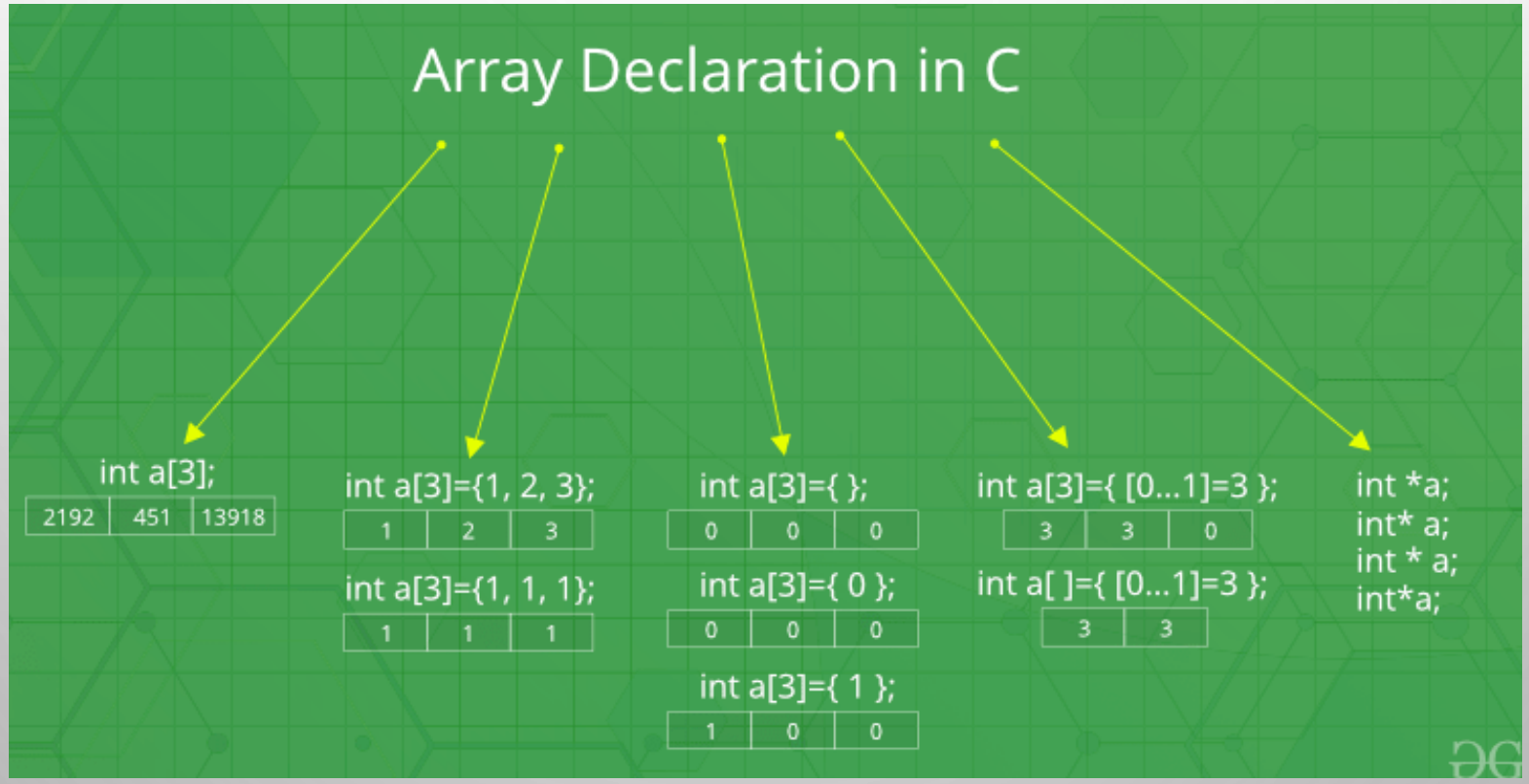
مبانی کامپیوتر و برنامه‌سازی

آرایه ها

- آرایه یک بعدی
- رشته
- آرایه دو بعدی



تعریف آرایه

- مجموعه‌ای از مقادیر مرتبط که همه یک نوع داده‌ای دارند
- دسترسی: با یک نام و شماره مکان
- شماره گذاری: شروع از صفر
- نام آرایه: تابع قوانین نام‌گذاری شناسه‌ها
  - ترکیبی از حروف و ارقام که با رقم شروع نشود

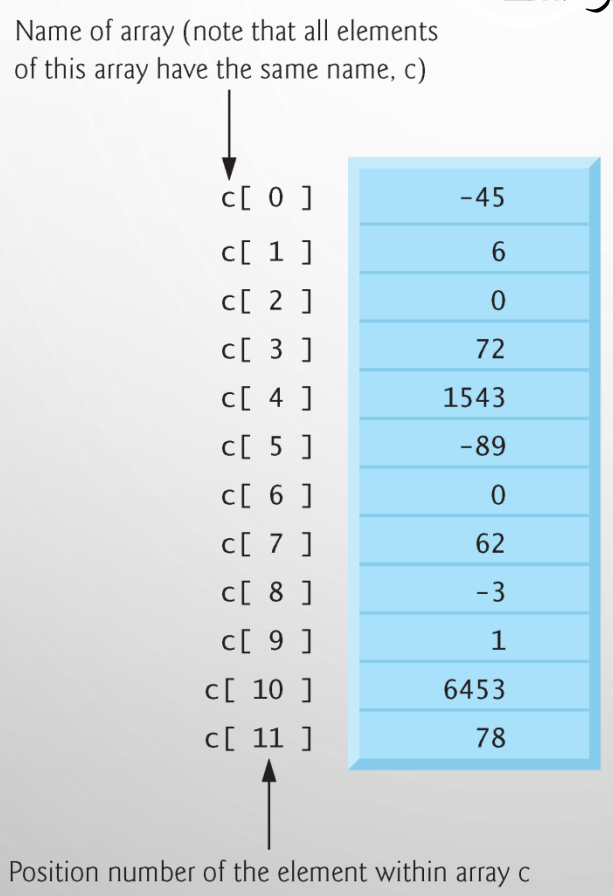
[شماره مکان]نام آرایه :دسترسی

- شماره را اندیس نیز می‌نامیم
  - مقدار عددی ثابت
  - عبارت دارای مقدار عددی
- مثال: آرایه‌ای با هشت عنصر، نمایش:



مثال

- آرایه ای از اعداد
- این آرایه C نام دارد و شامل 12 عنصر است



دسترسی به عناصر آرایه

- A[5]=9;
- A[5]+=3;
- PRINTF(A[0]+A[1]+A[3]);

- مثال: اندیس: عبارت ریاضی دارای مقدار عددی

- INT X,Y;
- X=2*3;
- Y=6-5;
- A[X+Y]=1; معادل: A[7]=1;

آرایه، کروشه، اولویت عملگر

- اولویت کروشه هم سطح پرانتز است

Operators						Associativity	Type
[]	()					left to right	highest
++	--	!	(*type*)			right to left	unary
*	/	%				left to right	multiplicative
+	-					left to right	additive
<	<=	>	>=			left to right	relational
==	!=					left to right	equality
&&						left to right	logical AND
||						left to right	logical OR
?:						right to left	conditional
=	+=	-=	*=	/=	%=	right to left	assignment
,						left to right	comma

تعریف (اعلان) آرایه

نوع‌داده‌ای نام‌آرایه[تعداد عناصر];

مثال: int a[8];

مثال:
```
int b[ 100 ], x[ 27 ];
```

یک آرایه 100 تایی (با ظرفیت 100 عنصر) از اعداد صحیح به نام B و یک آرایه 27 تایی از اعداد صحیح به نام X تعریف می‌شود

آرایه، مقداردهی و نمایش مقدار عناصر

```
1  /* Fig. 6.3: fig06_03.c
2     initializing an array */
3  #include <stdio.h>
4
5  /* function main begins program execution */
6  int main( void )
7  {
8     int n[ 10 ]; /* n is an array of 10 integers */
9     int i; /* counter */
10
11    /* initialize elements of array n to 0 */
12    for ( i = 0; i < 10; i++ ) {
13       n[ i ] = 0; /* set element at location i to 0 */
14    } /* end for */
15
16    printf( "%s%13s\n", "Element", "Value" );
17
18    /* output contents of array n in tabular format */
19    for ( i = 0; i < 10; i++ ) {
20       printf( "%7d%13d\n", i, n[ i ] );
21    } /* end for */
22
23    return 0; /* indicates successful termination */
24 } /* end main */
```

```
Element         Value
      0             0
      1             0
      2             0
      3             0
      4             0
      5             0
      6             0
      7             0
      8             0
      9             0
```

مقداردهی اولیه آرایه

آرایه باید مقداردهی شود

در غیر اینصورت، عناصر «مقدار نامعلوم» خواهد داشت

- مقداردهی اولیه:

- int a[5]={9,11,15,25,35};
- int b[5]={0}; 5 عنصر با مقادیر صفر
- int c[5]={3}; 5 عنصر، اولی 3 و بقیه با مقادیر صفر
- int d[]={7,5,11,9}; شمارش خودکار مقادیر، آرایه 4 عنصری

- int n[3]={6,8,4,2}; خطای نحوی: سه عنصر و چهار مقدار

مقداردهی اولیه

- C

```c
/* Fig. 6.4: fig06_04.c
   Initializing an array with an initializer list */
#include <stdio.h>

/* function main begins program execution */
int main( void )
{
   /* use initializer list to initialize array n */
   int n[ 10 ] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };
   int i; /* counter */

   printf( "%s%13s\n", "Element", "Value" );

   /* output contents of array in tabular format */
   for ( i = 0; i < 10; i++ ) {
      printf( "%7d%13d\n", i, n[ i ] );
   } /* end for */

   return 0; /* indicates successful termination */
} /* end main */
```

```
Element         Value
      0            32
      1            27
      2            64
      3            18
      4            95
      5            14
      6            90
      7            70
      8            60
      9            37
```

آرایه

اعداد زوج و تعریف ثابت

```
1  /* Fig. 6.5: fig06_05.c
2     Initialize the elements of array s to the even integers from 2 to 20 */
3  #include <stdio.h>
4  #define SIZE 10 /* maximum size of array */
5
6  /* function main begins program execution */
7  int main( void )
8  {
9     /* symbolic constant SIZE can be used to specify array size */
10    int s[ SIZE ]; /* array s has SIZE elements */
11    int j; /* counter */
12
13    for ( j = 0; j < SIZE; j++ ) { /* set the values */
14       s[ j ] = 2 + 2 * j;
15    } /* end for */
16
17    printf( "%s%13s\n", "Element", "Value" );
18
19    /* output contents of array s in tabular format */
20    for ( j = 0; j < SIZE; j++ ) {
21       printf( "%7d%13d\n", j, s[ j ] );
22    } /* end for */
23
24    return 0; /* indicates successful termination */
25 } /* end main */
```

```
Element         Value
      0             2
      1             4
      2             6
      3             8
      4            10
      5            12
      6            14
      7            16
      8            18
      9            20
```

دستور #define

- #define SIZE 10
  - قبل از کامپایل برنامه، پیش‌پردازشگر همه عبارات SIZE را با مقدار 10 جایگزین می‌کند
  - دستورات «راهنمایی پیش‌پردازشگر»، در پایان، علامت ; ندارند
    - اضافه کردن ; در پایان #define SIZE 10 معمولا ایجاد خطا می‌کند
      - دلیل: با داشتن #define SIZE 10; همه عبارات SIZE با عبارت 10; جایگزین می شوند
        - نتیجه: در سطر شماره 10 خواهیم داشت: INT S[10;]; خطاست
  - ثابت، معمولا با حروف بزرگ انگلیسی نوشته می‌شود که با متغیرها اشتباه گرفته نشود
  - کاربرد ثابت: اعمال آسان‌تر تغییرات در برنامه
    - تغییر در یک نقطه از برنامه آسان‌تر از تغییر در سه نقطه است
      - احتمال فراموش کردن اعمال تغییرات در یکی از سه نقطه وجود دارد
    - مثال، 20 مقدار به جای 10 مقدار #define SIZE 20

آرایه

حاصل‌جمع مقادیر

```
1  /* Fig. 6.6: fig06_06.c
2     Compute the sum of the elements of the array */
3  #include <stdio.h>
4  #define SIZE 12
5
6  /* function main begins program execution */
7  int main( void )
8  {
9     /* use initializer list to initialize array */
10    int a[ SIZE ] = { 1, 3, 5, 4, 7, 2, 99, 16, 45, 67, 89, 45 };
11    int i; /* counter */
12    int total = 0; /* sum of array */
13
14    /* sum contents of array a */
15    for ( i = 0; i < SIZE; i++ ) {
16       total += a[ i ];
17    } /* end for */
18
19    printf( "Total of array element values is %d\n", total );
20    return 0; /* indicates successful termination */
21 } /* end main */
```

```
Total of array element values is 383
```

آرایه

نمایش مقادیر موجود در آرایه بصورت نمودار میله‌ای

هر عدد جداگانه چاپ شده و سپس به همان تعداد علامت ستاره «*» چاپ شود

```
Element         Value         Histogram
      0            19         ********************
      1             3         ***
      2            15         ***************
      3             7         *******
      4            11         ***********
      5             9         *********
      6            13         *************
      7             5         *****
      8            17         *****************
      9             1         *
```

نمایش نمودار میله‌ای

```
1  /* Fig. 6.8: fig06_08.c
2     Histogram printing program */
3  #include <stdio.h>
4  #define SIZE 10
5
6  /* function main begins program execution */
7  int main( void )
8  {
9     /* use initializer list to initialize array n */
10    int n[ SIZE ] = { 19, 3, 15, 7, 11, 9, 13, 5, 17, 1 };
11    int i; /* outer for counter for array elements */
12    int j; /* inner for counter counts *s in each histogram bar */
13
14    printf( "%s%13s%17s\n", "Element", "Value", "Histogram" );
15
16    /* for each element of array n, output a bar of the histogram */
17    for ( i = 0; i < SIZE; i++ ) {
18       printf( "%7d%13d          ", i, n[ i ]) ;
19
20       for ( j = 1; j <= n[ i ]; j++ ) { /* print one bar */
21          printf( "%c", '*' );
22       } /* end inner for */
23
24        printf( "\n" ); /* end a histogram bar */
25    } /* end outer for */
26
27    return 0; /* indicates successful termination */
28 } /* end main */
```

آرایه

بزرگترین عنصر در مکان شماره صفر قرار گیرد

```c
#include <stdio.h>
int main(){
     int i,n;
     float arr[100];
     printf("Enter total number of elements(1 to 100): ");
     scanf("%d",&n);
     printf("\n");
     for(i=0;i<n;++i)   /* Stores number entered by user. */
     {
         printf("Enter Number %d: ",i+1);
         scanf("%f",&arr[i]);
     }
     for(i=1;i<n;++i)   /* Loop to store largest number to arr[0] */
     {
         if(arr[0]<arr[i]) /* Change < to > if you want to find smallest element
              arr[0]=arr[i];
     }
     printf("Largest element = %.2f",arr[0]);
     return 0;
}
```

آرایه

- میانگین مقادیر عناصر آرایه

```c
#include <stdio.h>
int main(){
     int n, i;
     float num[100], sum=0.0, average;
     printf("Enter the numbers of data: ");
     scanf("%d",&n);
     while (n>100 || n<=0)
     {
          printf("Error! number should in range of (1 to 100).\n");
          printf("Enter the number again: ");
          scanf("%d",&n);
     }
    for(i=0; i<n; ++i)
    {
        printf("%d. Enter number: ",i+1);
        scanf("%f",&num[i]);
        sum+=num[i];
    }
    average=sum/n;
    printf("Average = %.2f",average);
    return 0;
}
```

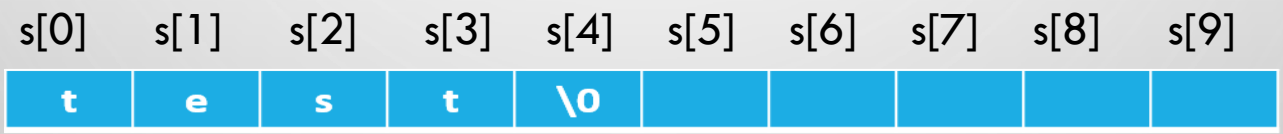
```
Enter the numbers of data: 6
1. Enter number: 45.3
2. Enter number: 67.5
3. Enter number: -45.6
4. Enter number: 20.34
5. Enter number: 33
6. Enter number: 45.6
Average = 27.69
```

آرایه رشته‌ای

- رشته آرایه‌ای از کاراکترها است
- انتهای رشته بوسیله کاراکتر خاصی به نام (تهی=NULL) که با استفاده از ’\0‘ ایجاد می‌شود، مشخص می‌شود
- اگر رشته را به صورت زیر تعریف کنیم فقط از 9 کاراکتر آن می توان استفاده کرد

```
char s[10];
```

- اگر محتویات این رشته برابر عبارت test شود، این رشته به صورت زیر نشان داده می‌شود

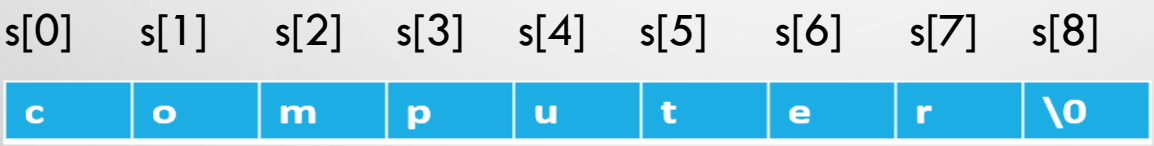


- طول رشته را باید یک واحد بیشتر از آنچه نیاز داریم، تعیین کنیم (کاراکتر \0)

مقداردهی اولیه به رشته‌ها

- دو روش وجود دارد
  - رشته در داخل دابل کوتیشن قرار گیرد و به متغیر رشته‌ای نسبت داده شود
  - هر یک از کاراکترهای رشته ای به عنوان یک عنصر آرایه نسبت داده شود

```
char s[]="computer";
char st[11]="algorithm";
char str[]={'p', 'r', 'g', '\0'};
```







رشته

ورودی و خروجی رشته‌ها

```c
#include <stdio.h>
int main(){
     char name[20];
     printf("Enter name: ");
     scanf("%s",name);
     printf("Your name is %s.",name);
     return 0;
}
```

Output

```
Enter name: Dennis Ritchie
Your name is Dennis.
```

رشته

ورودی و خروجی رشته‌ها

```
int main(){
     char name[30];
     printf("Enter name: ");
     gets(name);       //Function to read string from user.
     printf("Name: ");
     puts(name);      //Function to display string.
     return 0;
}
```

Output

```
Enter name: Tom Hanks
Name: Tom Hanks
```

رشته

- جستجو و شمارش کاراکترها

```c
#include <stdio.h>
int main(){
    char c[1000],ch;
    int i,count=0;
    printf("Enter a string: ");
    gets(c);
    printf("Enter a character to find frequency: ");
    scanf("%c",&ch);
    for(i=0;c[i]!='\0';++i)
    {
         if(ch==c[i])
              ++count;
    }
    printf("Frequency of %c = %d", ch, count);
    return 0;
}
```

Output

```
Enter a string: This website is awesome.
Enter a character to find frequency: e
Frequency of e = 4
```

رشته

- کپی، الحاق و طول رشته

```c
#include <stdio.h>
#include <string.h>

int main () {

   char str1[12] = "Hello";
   char str2[12] = "World";
   char str3[12];
   int  len ;

   /* copy str1 into str3 */
   strcpy(str3, str1);
   printf("strcpy( str3, str1) :  %s\n", str3 );

   /* concatenates str1 and str2 */
   strcat( str1, str2);
   printf("strcat( str1, str2):   %s\n", str1 );

   /* total lenghth of str1 after concatenation */
   len = strlen(str1);
   printf("strlen(str1) :  %d\n", len );

   return 0;
}
```

```
strcpy( str3, str1) :  Hello
strcat( str1, str2):   HelloWorld
strlen(str1) :  10
```

رشته

مقایسه، strcmp

```c
#include <stdio.h>
#include <string.h>
int main()
{
     char s1[20] = "BeginnersBook";
     char s2[20] = "BeginnersBook.COM";
     if (strcmp(s1, s2) ==0)
     {
        printf("string 1 and string 2 are equal");
     }else
      {
         printf("string 1 and 2 are different");
      }
     return 0;
}
```

```
string 1 and 2 are different
```

رشته

دستورات مربوط عملیات رشته ای

Function & Purpose
strcpy(s1, s2); Copies string s2 into string s1.
strcat(s1, s2); Concatenates string s2 onto the end of string s1.
strlen(s1); Returns the length of string s1.
strcmp(s1, s2); Returns 0 if s1 and s2 are the same; less than 0 if s1<s2; greater than 0 if s1>s2.

آرایه

آرایه دوبعدی

```
float a[2][6];
```

	col 1	col 2	col 3	col 4	col 5	col 6
row 1	a[0][0]	a[0][1]	a[0][2]	a[0][3]	a[0][4]	a[0][5]
row 2	a[1][0]	a[1][1]	a[1][2]	a[1][3]	a[1][4]	a[1][5]

Figure: Multidimensional Arrays

آرایه، مقداردهی اولیه

آرایه دوبعدی، مقداردهی اولیه

```
int c[2][3]={{1,3,0}, {-1,5,9}};
                      OR
int c[][3]={{1,3,0}, {-1,5,9}};
                      OR
int c[2][3]={1,3,0,-1,5,9};
```

آرایه سه‌بعدی: مقداردهی اولیه

```
double cprogram[3][2][4]={
{{-0.1, 0.22, 0.3, 4.3}, {2.3, 4.7, -0.9, 2}},
 {{0.9, 3.6, 4.5, 4}, {1.2, 2.4, 0.22, -1}},
 {{8.2, 3.12, 34.2, 0.1}, {2.1, 3.2, 4.3, -2.0}}
};
```

آرایه دوبعدی، جمع دو ماتریس

تت

```c
#include <stdio.h>
int main(){
    float a[2][2], b[2][2], c[2][2];
    int i,j;
    printf("Enter the elements of 1st matrix\n");
/* Reading two dimensional Array with the help of two for loop. If there was a
    for(i=0;i<2;++i)
         for(j=0;j<2;++j){
         printf("Enter a%d%d: ",i+1,j+1);
         scanf("%f",&a[i][j]);
         }
    printf("Enter the elements of 2nd matrix\n");
    for(i=0;i<2;++i)
         for(j=0;j<2;++j){
         printf("Enter b%d%d: ",i+1,j+1);
         scanf("%f",&b[i][j]);
         }
    for(i=0;i<2;++i)
         for(j=0;j<2;++j){
/* Writing the elements of multidimensional array using loop. */
         c[i][j]=a[i][j]+b[i][j];  /* Sum of corresponding elements of two
         }
    printf("\nSum Of Matrix:");
    for(i=0;i<2;++i)
         for(j=0;j<2;++j){
         printf("%.1f\t",c[i][j]);
              if(j==1)                /* To display matrix sum in order.
                  printf("\n");
        }
return 0;
}
```

```
Enter the elements of 1st matrix
Enter a11: 2;
Enter a12: 0.5;
Enter a21: -1.1;
Enter a22: 2;
Enter the elements of 2nd matrix
Enter b11: 0.2;
Enter b12: 0;
Enter b21: 0.23;
Enter b22: 23;

Sum Of Matrix:
2.2       0.5
-0.9      25.0
```

آرایه، جستجوی خطی

search

```c
#include<stdio.h>
int main(){

    int a[10],i,n,m,c=0;

    printf("Enter the size of an array: ");
    scanf("%d",&n);

    printf("Enter the elements of the array: ");
    for(i=0;i<=n-1;i++){
        scanf("%d",&a[i]);
    }

    printf("Enter the number to be search: ");
    scanf("%d",&m);
    for(i=0;i<=n-1;i++){
        if(a[i]==m){
            c=1;
            break;
        }
    }
    if(c==0)
        printf("The number is not in the list");
    else
        printf("The number is found");

    return 0;
}
```

Sample output:

```
Enter the size of an array: 5
Enter the elements of the array: 4 6 8 0 3
Enter the number to be search: 0
The number is found
```