

فصل چهارم

اعداد تصادفی



I. ASHKAVAND RAD

فهرست مطالب فصل چهارم

۱. تولید اعداد تصادفی
۲. تعریف نوع داده (typedef)
۳. داده های از نوع شمارشی
۴. فرمت های مختلفه مقادیر خروجی



اعداد تصادفی

مقادیر تصادفی یا شانسی در اکثر برنامه‌های کاربردی در زمینه شبیه‌سازی و بازیهای کامپیوتری نقش مهمی را ایفا می‌نمایند. برای ایجاد یک عدد تصادفی صحیح بین ۰ و ۳۲۷۶۷ با استفاده از تابع rand () استفاده نمائیم.

rand()



برنامه زیر 10 عدد تصادفی بین 0 و 32767 را ایجاد می نماید.

```
#include      <stdlib.h>
#include      <iostream.h>
int main( )
{
for(int j=1; j<=10; ++j)
cout << rand( ) << '\n' ;
return 0 ;
}
```



نکته :

اگر برنامه فوق را چندبار اجرا نمائیم جواب یکسانی را از کامپیوتر می گیریم.
برای تصادفی کردن اعداد می بایستی از تابع `srand()` استفاده نمائیم.
این تابع به یک آرگومان صحیح از نوع `unsigned` نیاز دارد.
به این آرگومان `seed` گفته می شود.

در اسلاید بعد برنامه قبلی را با تابع `srand()` نوشته ایم.



برنامه زیر 10 عدد تصادفی بین 0 و 32767 را ایجاد می نماید. (srand())

```
#include      <stdlib.h>
#include      <iostream.h>
int main( )
{
  unsigned   seed;
  cout << "Enter seed value : " ;
  cin >> seed ;
  srand(seed);
  for(int j=1; j<=10; ++j)
  cout << rand( ) << ' \n ' ;
  return 0 ;
}
```



برنامه زیر نتیجه پرتاب دو تاس را نمایش می دهد.

```
#include <iostream.h>
#include <stdlib.h>
int main( )
{
    unsigned seed, d1, d2;
    cout << "Enter seed: " ;
    cin >> seed ;
    srand(seed) ;
    d1= 1+rand( )% 6 ;
    d2= 1+rand( )% 6 ;
    cout << d1 << " " << d2 ;
    return 0 ;
}
```



برنامه زیر 10 اعداد شانسی بین 0 و 1 را نمایش می دهد.

```
#include <stdlib.h>
#include <iostream.h>
int main( )
{
    unsigned seed ;
    cout << "Enter seed: " ;
    cin >> seed ;
    srand(seed) ;
    for(int i=1; i<=10; ++i)
    cout << rand( ) / 32768.0 << endl ;
    return 0 ;
}
```



I . ASHKAVAND RAD

تعریف نوع داده (typedef)

از **typedef** می‌توان برای تعریف نوع داده‌های جدید که معادل نوع داده‌های موجود باشد استفاده نمود. شکل کلی عبارتست از:

```
typedef type newtype;
```

نشاندنده نوع داده موجود

اسم جدید



مثال:

```
typedef int integer;
```

حال می توان x و y را بصورت زیر تعریف نمود:

```
integer x,y;
```



داده‌های از نوع شمارشی

بمنظور معرفی داده‌های از نوع شمارشی از کلمه `enum` استفاده می‌گردد.

مثال:

```
enum color {red, blue, green, yellow, brown} ;
```

`color` یک نوع داده شمارشی می‌باشد.



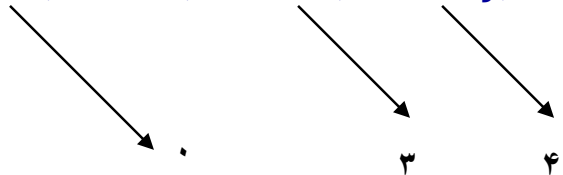
چند مثال :

```
enum status {married, divorced, vidow, single};  
status a ;  
a= single ;
```

```
enum days {sat, sun, mon, tue, wed, thr,  
           fri};
```

```
enum bread {lavash, fantezi, taftoon, barbari};
```

```
enum color { yellow, red=2, brown, white };  
color x=brown;
```



توجه :

بایستی در نظر داشت که داده‌های از نوع شمارشی در عملیات ورودی و خروجی شرکت نمی‌نمایند. عبارت دیگر مقادیر داده‌های از نوع شمارشی بایستی در برنامه تعیین نمود. دستورالعمل‌های ورودی و خروجی مانند **cin** و **cout** در مورد داده‌های شمارشی نمی‌توان استفاده نمود.



فرمتهای مختلف مقادیر خروجی

مقدار X بطور غیر علمی با نقطه اعشار ثابت نمایش داده می شود.

```
include <iomanip.h>
```

```
double x=1050 ;
```

```
cout << setiosflags(ios :: fixed | ios :: showpoint ) << setw(23)  
<< setprecision(2) << x << endl ;
```

مقدار X با دو رقم اعشار نمایش داده می شود.

مقدار X با طول میدان ۲۳ نمایش داده می شود.

بنابراین مقدار X بصورت زیر نمایش داده می شود :

1050.00

I . ASHKAVAND RAD

