

# فصل هفتم: «حلقه و تکرار»

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 ی دستیابی به آنها
  - ۱۱ کار با فایل ها
  - ۱۲ اشاره گر ها
  - ۱۳ ساختار ها و یونیون ها
- اهمیتی

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ قاتمه حلقه ۶ پرش ۷ اعداد شبه تصادفی ۸

## مقدمه

**تکرار (repetition):** اجرای پی در پی یک دستور یا بلوکی از دستورالعمل ها در یک برنامه؛ اجبار کنترل برنامه برای بازگشت به خطوط قبلی و اجرای دوباره آنها

طبیعت چرخه مانندشان: **حلقه (loop)**؛ مجموعه ای از چند دستور که تا زمانی که شرط ادامه حلقه (loop-continuation condition) درست است کامپیوتر آنها را به طور مکرر اجرا می کند. C++ دارای سه دستور تکرار است:



while 1  
do\_while 2  
for 3

3/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ قاتمه حلقه ۶ پرش ۷ اعداد شبه تصادفی ۸

## دو حالت تکرار:

### ۱- تکرار کنترل شده توسط شمارنده، تکرار معین (definite repetition)

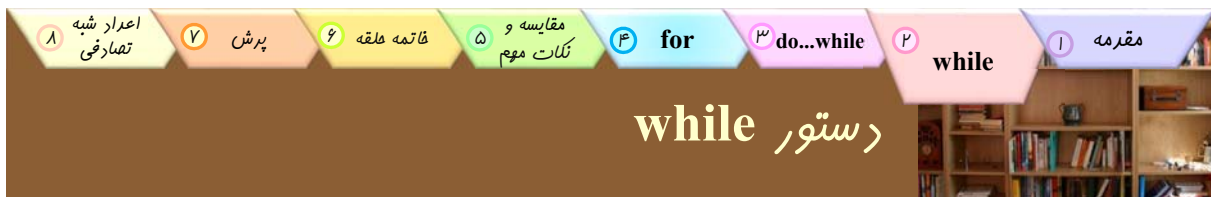
برای شمارش تعداد دفعات تکرار: یک **متغیر کنترلی (control variable)** که با هر بار اجرای گروهی از دستورها (معمولاً ۱ واحد) افزایش می یابد. هنگامی که مقدار متغیر کنترلی تعیین نماید تعداد درستی از تکرارها انجام شده، حلقه قاتمه می یابد و کامپیوتر با دستوری که بعد از دستور تکرار است به اجرا ادامه می دهد.



### ۲- تکرار کنترل شده توسط نگهبان، تکرار نامعین (indefinite repetition)

- ۱- تعداد دقیق تکرار از قبل معلوم نباشد،
- ۲- حلقه شامل دستورهائی است که در هر بار اجرای حلقه، داده ای را دریافت می کند. پایان داده ها توسط مقدار نگهبان مشخص می گردد.





## دستور while

**هدف:** تکرار یک فرآیند تا وقتی که شرطی برقرار است.

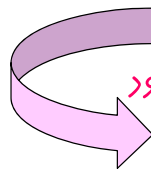
**نمونه استفاده:**

**while (condition) statement;**

**condition:** شرط

**Statement:** دستوری که باید تکرار شود.

عبارت درون پرانتز جهت تصمیم گیری ارزیابی می شود،



**پنانه شرط برقرار بود**

دستوری که به دنبال پرانتز آورده شده است اجرا می گردد.

پس از آن دوباره همان عبارت ارزیابی می شود و در صورت برقرار بودن شرط اجرای دستور تکرار می گردد. این عمل تا زمانی که شرط برقرار است ادامه



مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

## مثال ۱) محاسبه حاصل جمع اعداد صحیح متوالی با حلقه while

این برنامه مقدار  $1+2+3+...+n$  را برای عدد ورودی  $n$  محاسبه می کند:

```
int main()
{
    int n, i=1;
    cout << "Enter a positive integer:";
    cin >> n;
    long sum=0;
    while (i <= n)
        sum += i++;
    cout << "The sum of the first " << n << " integers is " << sum << endl;
}
```

Enter a positive integer: 8

The sum of the first 8 integers is 36

i	1	2	3	4	5	6	7	8	9
sum	0	1	3	6	10	15	21	28	36

Enter a positive integer: 100

The sum of the first 100 integers is 5050

7/47

13/11/2016

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

## مثال ۲) استفاده از حلقه while برای تکرار یک محاسبه

```
int main ()
{
    double x;
    cout << "Enter a positive number: ";
    cin >> x;
    while (x > 0)
    {
        cout << "sqrt(" << x << ") = " << sqrt(x) << endl;
        cout << "Enter another positive number (or 0 to quit): ";
        cin >> x;
    }
    return 0;
}
```

برنامه ای بنویسید که جزر هر عددی که کاربر وارد کند را محاسبه می نماید و کاربر مجبور نباشد برای محاسبه جزر عدد بعدی، برنامه را دوباره اجرا کند:

**X:** متغیر نگهبان حلقه یا کنترل حلقه

Enter a positive number: 36

sqrt(36) = 6

Enter another positive number (or 0 to quit): 84

sqrt(84) = 9.161551

Enter another positive number (or 0 to quit): 0

8/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016



مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

## دستور do..while

هدف: ساختن حلقه

نمونه استفاده:

do statement while (condition);

condition: شرط؛

statement: دستور یا بلوکی که تکرار است تکرار شود.

این دستور ابتدا statement را اجرا می کند و سپس شرط condition را بررسی می کند. اگر شرط درست بود حلقه دوباره تکرار می شود وگرنه حلقه پایان می یابد. ابتدا، دستوری که پس از کلمه کلیدی do آورده شده و بدنه حلقه تکرار را تشکیل می دهد، اجرا می گردد. سپس عبارت درون پرانتز جهت تصمیم گیری ارزیابی می شود. اگر معادل true بود (شرط برقرار بود)، بدنه دو مرتبه اجرا می شود و باز همان عبارت ارزیابی می گردد. این عمل تا زمانی که شرط برقرار است ادامه می یابد. همین که جواب عبارت معادل false شود، دیگر اجرای دستور تکرار نمی شود و سراغ دستورهای بعدی می رود.

9/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016



مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

مثال ۳) مناسبه حاصل جمع اعداد صحیح متوالی با حلقه do.. While

```

int main()
{
    int n, i=0;
    cout << "Enter a positive integer: ";
    cin >> n;
    long sum=0;
    do
        sum += i++;
    while (i <= n);
    cout << "The sum of the first " << n << " integers is " << sum << endl;
    return 0;
}

```

```

{ int n, i=1;

```

```

while (i <= n)
    sum += i++;
cout << ...
}

```



مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

مثال ۴) برنامه‌ای بنویسید که همه اعداد فاکتوریل را که از عدد داده شده کوچک ترند، چاپ کند.

اعداد فاکتوریل  $0!, 1!, 2!, 3!$  و ... با استفاده از رابطه‌های بازگشتی زیر تعریف می‌شوند:

$$0! = 1, \quad n! = n(n-1)!$$

برای مثال به ازای  $n=1$  در معادله دوم داریم:

$$1! = 1((1-1)!) = 1(0!) = 1(1) = 1$$

همچنین برای  $n=2$  داریم:

$$2! = 2((2-1)!) = 2(1!) = 2(1) = 2$$

و به ازای  $n=3$  داریم:

$$3! = 3((3-1)!) = 3(2!) = 3(2) = 6$$



```
int main()
```

```
{ long long unsigned bound;
```

```
  cout << "Enter a positive integer: ";
```

```
  cin >> bound;
```

```
  cout << "Factorial numbers < " << bound << ":\n1";
```

```
  long f=1, i=1;
```

```
  do
```

```
  { cout << ", " << f;
```

```
    f *= ++i;
```

```
  }
```

```
  while (f < bound);
```

```
  cout << endl;
```

```
  return 0;
```

```
}
```

13/47



Enter a positive integer: 180

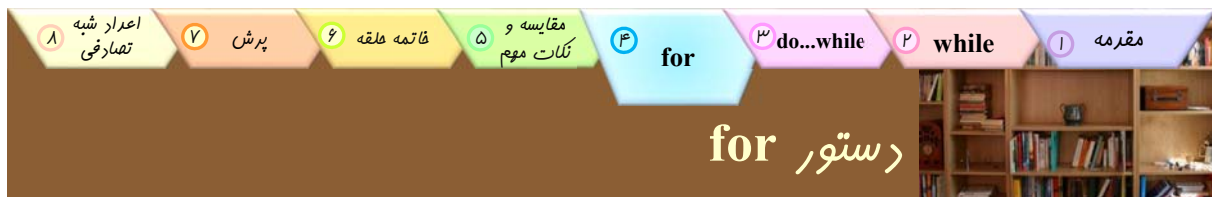
Factorial numbers < 180:

1, 1, 2, 6, 24, 120



## مقایسه while و do...while

از آنجا که در دستور **while** ابتدا شرط بررسی می شود، گاهی ممکن است حلقه اصلا اجرا نگردد. در حالی که در **do...while** حداقل یک مرتبه بدنه دستور اجرا می گردد، سپس جهت ادامه اجرای حلقه تکرار تصمیم گیری می شود. بعلاوه در **do...while** متغیر کنترلی به جای این که قبل از شروع حلقه تنظیم شود، می تواند درون آن تنظیم گردد.



نحوه استفاده:

**for (initialization; condition; update) statement;**

سه قسمت داخل پرانتز، حلقه را کنترل می کنند.

**Initialization:** برای اعلان یا مقداردهی اولیه به متغیر کنترل حلقه

اولین عبارتی که ارزیابی می شود پیش از این که نوبت به تکرارها برسد.

**Condition:** برای تعیین این که آیا حلقه باید تکرار شود یا خیر. شرط کنترل

حلقه: درست باشد ← دستور statement اجرا می شود.

**Update:** برای پیش بردن متغیر کنترل حلقه. پس از اجرای statement

ارزیابی می گردد.



ترتیب مراحلی که در دستور **for** انجام می شود:

۱- ارزیابی عبارت initialization

۲- بررسی شرط **condition** اگر نادرست باشد، حلقه فاصله می یابد.

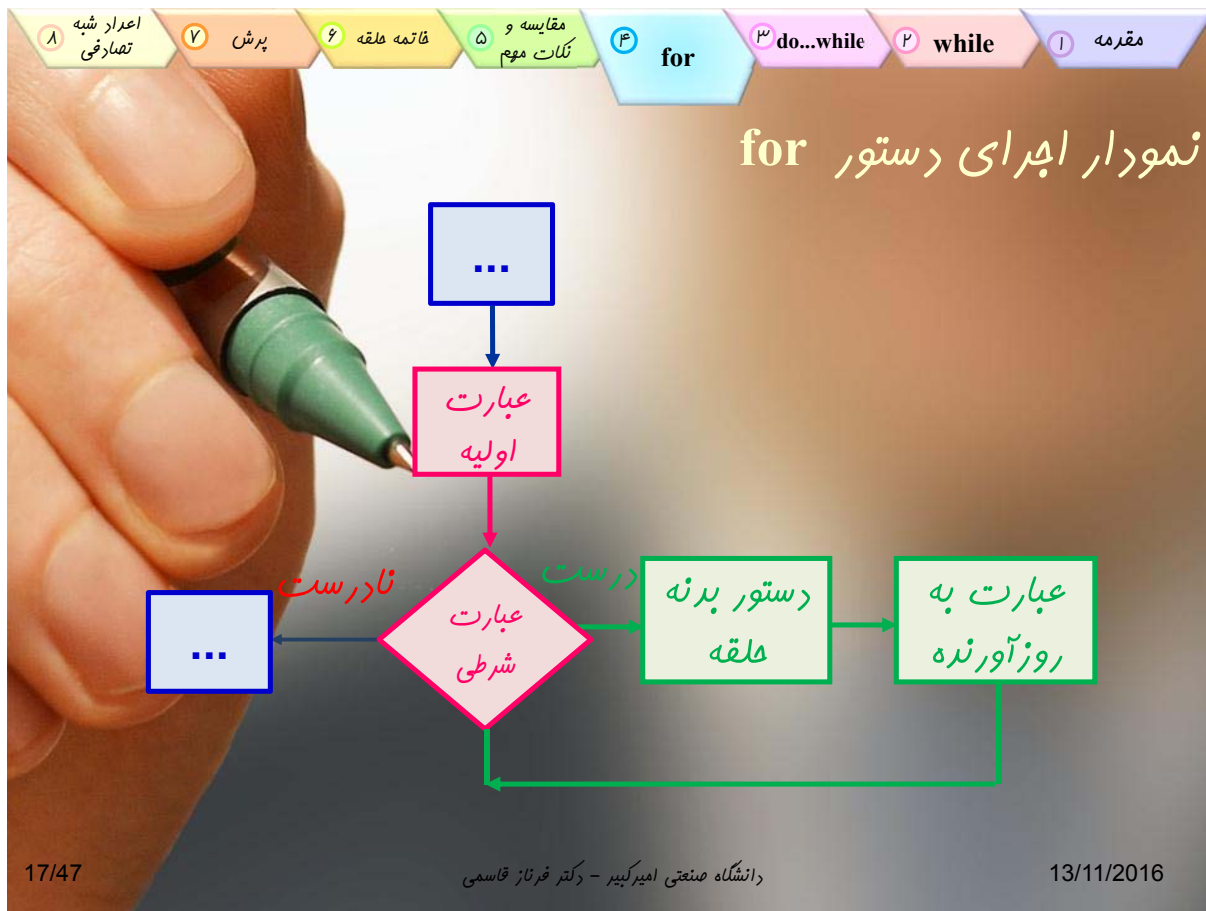
۳- اگر **condition** درست باشد، اجرای statement

۴- ارزیابی عبارت **update**

۵- تکرار گام های ۲ تا ۴

عبارت های initialization و condition و update عبارت های اختیاری هستند. یعنی می توانیم آنها را در حلقه ذکر نکنیم.





مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

## مثال ۵) استفاده از حلقه for برای محاسبه مجموع اعداد صحیح متوالی

```

int main()
{
    int n;
    cout << "Enter a positive integer: ";
    cin >> n;
    long sum=0;
    for (int i=1; i <= n; i++)
        sum += i;
    cout << "The sum of the first " << n << " integers is " << sum << endl;
    return 0;
}
  
```

Enter a positive integer: 7  
The sum of the first 8 integers is 28

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

مثال ۶) برنامه بنویسید که ده عدد صحیح مثبت را به ترتیب نزولی چاپ کند.

```
int main()
{ for (int i=10; i > 0; i--)
  cout << " " << i;
  cout << endl;
}
```

یک حلقه for نزولی



10 9 8 7 6 5 4 3 2 1

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

مثال ۷) برنامه ای بنویسید که همه اعداد فاکتوریل را که از عدد داده شده کوچک ترند، چاپ کند.

```
int main()
{ long bound;
  cout << "Enter a positive integer: ";
  cin >> bound;
  cout << "Factorial numbers < " << bound << ":\n1";
  long f=1;
  for (int i=2; f <= bound; i++)
  { cout << ", " << f;
    f *= i;
  }
  cout<<endl;
}
```



دستور for انعطاف پذیری بیشتری به برنامه می دهد

Enter a positive integer: 100000

Factorial numbers < 100000:

1, 1, 2, 6, 24, 120, 720, 5040, 40320, 362880

مقدمه ۱ while ۲ do...while ۳ for ۴ مقایسه و نکات مهم ۵ قاتمه حلقه ۶ پرش ۷ اعداد شبه تصادفی ۸

## مقایسه‌ی while و for

از نظر منطقی دستور زیر:

```
for ( exp1; exp2; exp3 )
    statement
```

معادل است با:

```
exp1;
while (exp2 )
{
    statement
    exp3;
}
```

برای انتقال دستور مناسب، باید  
به نوع اجرا و این که کدام روش،  
فرآیند مورد نظر را واضح تر ارائه  
می دهد، توجه کرد.

و از نگاه دیگر:

```
while (exp )
    statement
```

از نظر منطقی معادل است با:

```
for ( ; exp ; )
    statement
```

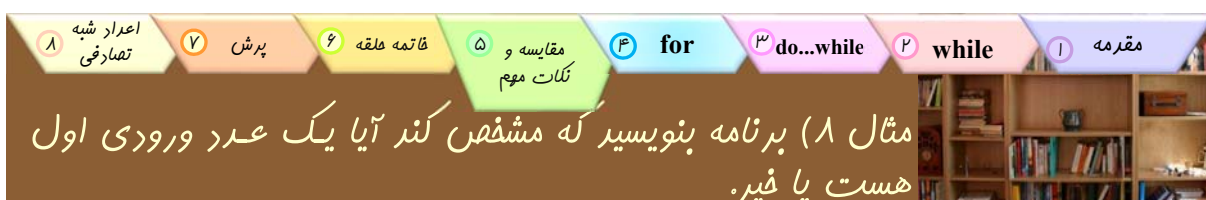
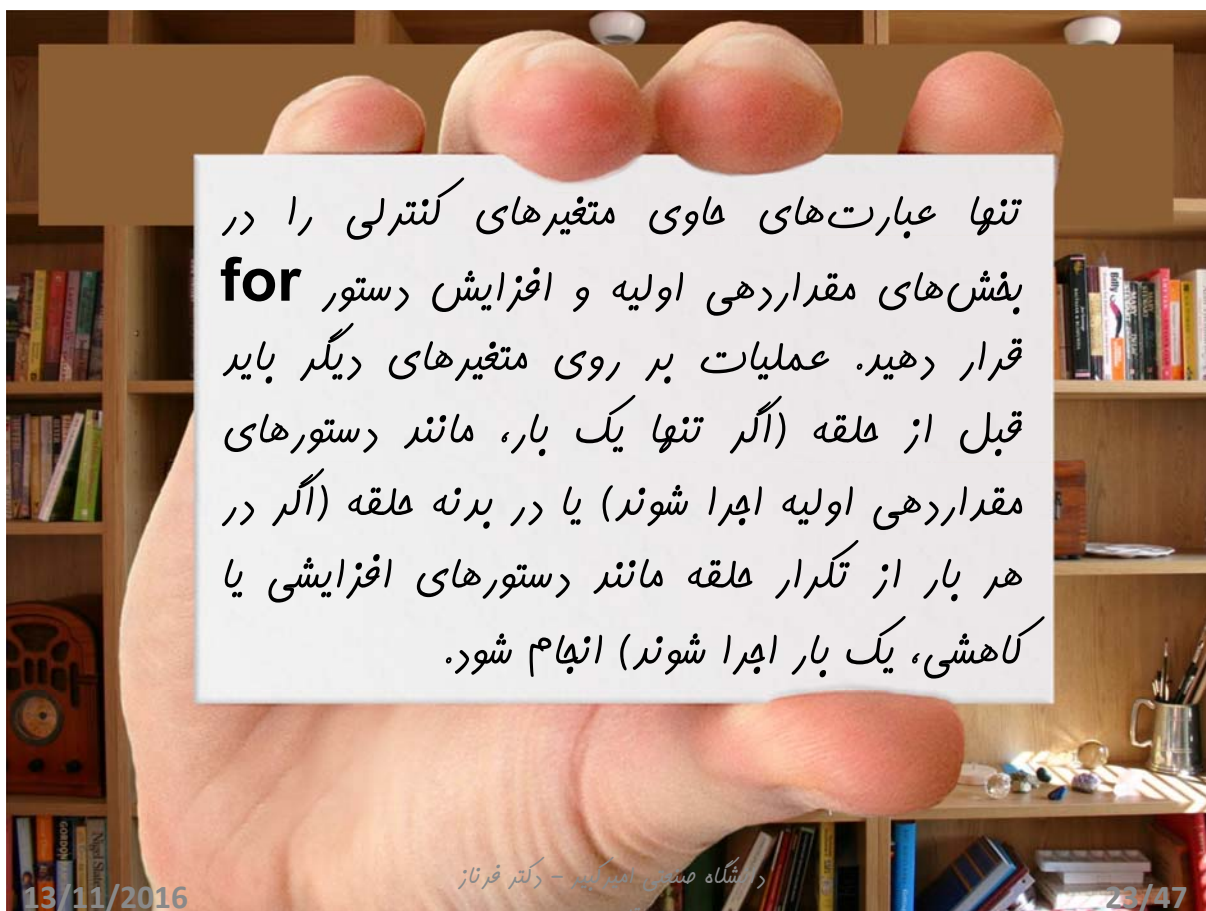
۱- شمارش حلقه ها (شمارنده حلقه) را با مقادیر صحیح کنترل کنید و از مقادیر اعشاری برای کنترل شمارنده حلقه استفاده نکنید.

۲- دستورهای داخل بدنه هر دستور کنترلی را به صورت تو رفته بنویسید.

13/11/2016

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

22/47



```
int main()
{ long n; bool f=1;
  cout << "Enter a positive integer: ";
  cin >> n;
  if (n < 2) cout << n << " is not prime." << endl;
  else if (n < 4) cout << n << " is prime." << endl;
  else if (n%2 == 0) cout << n << " = 2*" << n/2 << endl;
  else
  { for (int d=3; f && (d <= sqrt(n)); d+=2)
    { if (n%d == 0)
      { cout << n << " = " << d << "*" << n/d << endl;
        f=0; //exit(0);
      }
    }
    if (f) cout << n << " is prime." << endl;
  }
}
```

استفاده از حلقه **for** با گام‌های بزرگتر از یک



Enter a positive integer: 101

101 is prime.

Enter a positive integer: 975313579

975313579 = 17\*57371387



مقدمه ۱ while ۲ do...while ۳ for ۴ مقایسه و نکات مهم ۵ قاتمه حلقه ۶ پرش ۷ اعداد شبه تصادفی ۸

مثال ۹) بیک اختلاف بین بزرگترین و کوچکترین عدد از یک لیست اعداد را بیابد:

استفاده از نگهدارنده برای کنترل حلقه for

```
int main()
{
    int i, min, Max, x, n;
    cin>>n;
    cin>>Max;
    min=Max;
    for (i=1;i<n;i++)
    {
        cin>>x;
        if (x>Max) Max=x;
        else if (x<min) min=x;
    }
    cout<<"Max-min:"<<Max<<" - "<<min<<" = "<<Max-min<<endl;
    getch();
    return 0;
}
```

بیک تفاضل دو عدد بزرگ یک لیست n تایی از اعداد را بیابد.

25/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016

مقدمه ۱ while ۲ do...while ۳ for ۴ مقایسه و نکات مهم ۵ قاتمه حلقه ۶ پرش ۷ اعداد شبه تصادفی ۸

مثال ۱۰) برنامه ای بنویسید که دو عدد را گرفته و ب.م.م آنها را مناسبه نماید.

```
#include <iostream>
#include <conio.h>
using namespace std;
int main ( )
{
    int a, b, c = 1;
    cout<<"Enter a, b to compute gcd(a,b):";
    cin>>a>>b;
    while (c!=0)
    {
        c = a % b;
        a = b;
        b = c;
    }
    cout<<"gcd="<<a;
    getch();
    return 0;
}
```

Enter a, b to compute gcd(a,b):45 5  
gcd=5



26/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016



اعداد شبه تصادفی ۸ | پرش ۷ | قاتمه حلقه ۶ | مقایسه و نکات مهم ۵ | for ۴ | do...while ۳ | while ۲ | مکرره ۱

مثال ۱۱) برنامه‌ای بنویسید که با فشردن هر دکمه روی صفحه کلید توسط کاربر کداسکی آن را نمایش دهد و در صورتی که **q** وارد شود، از برنامه خارج شود.

Press q. to exit

You have pressed r and its code is code=114

You have pressed R and its code is code=82

You have pressed S and its code is code=53

You have pressed a and its code is code=97

You have pressed A and its code is code=65

You have pressed q and its code is code=113

Press and key to continue ...

```
#include <iostream>
#include <conio.h>
using namespace std;
int main ()
{char ch=0 ;
cout<<"press q, to exit\n";
while (ch!= 'q')
{
    ch = getch( );
    cout<<"you have pressed "<<ch<<" and its code=" <<(int) (ch) <<endl;
}
    return 0;
}
```

اعداد شبه تصادفی ۸ | پرش ۷ | قاتمه حلقه ۶ | مقایسه و نکات مهم ۵ | for ۴ | do...while ۳ | while ۲ | مکرره ۱

مثال ۱۲) برنامه‌ای بنویسید که یک عدد صحیح از ورودی خوانده و ارقام آن را به صورت جداگانه (تجزیه شده) بر روی صفحه چاپ کند.

```
int main ()
{
    int k;
    cout<< "please Enter an integer: ";
    cin>>k;
    while (k>0)
    {
        cout<<k%10<<" ";
        k=k/10;
    }
}

int main ()
{
    int k,n,i=0, s=1;
    cout<< "please Enter an integer: ";
    cin>>n;
    k=n;
    while (k>0)
    {
        k=k/10;
        i++;
        s*=10;
    }
    k=n;
    for (int j=i-1;j>0; --j)
    {
        s/=10;
        cout<<k/(s)<<"t";
        k=k%(s);
    }
    cout<< k << endl;
}
```

مقدمه ۱ while ۲ do...while ۳ for ۴ مقایسه و نکات مهم ۵ قاتمه حلقه ۶ پرش ۷ اعداد شبه تصادفی ۸

مثال ۱۳) برنامه‌ای بنویسید که  $n$  را از کاربر گرفته و عدد  $\pi$  را مطابق رابطه تا  $n$  جمله مناسبه نماید.

```
int main ( )
{
    int n, i;
    double p=0;
    cout<<"Enter: ";
    cin>>n;
    for (i =0 ; i < n ; i++)
    {
        if (i % 2 == 0)
            p +=1.0 /(2*i+1);
        else
            p += -1.0 / (2*i+1);
    }
    cout<<4*p;
    getch ( );
    return 0;
}
```

جمله عمومی سری  $\frac{(-1)^i}{2i+1}$

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} \dots$$

9/47

مقدمه ۱ while ۲ do...while ۳ for ۴ مقایسه و نکات مهم ۵ قاتمه حلقه ۶ پرش ۷ اعداد شبه تصادفی ۸

عملگر کاما (,)

در محل عبارت اولیه در دستور for می‌توان بیش از یک عبارت را بکار برد. برای این منظور باید میان عبارت‌ها عملگر کاما (,) قرار گیرد. معمولاً در زمانی از این امکان استفاده می‌شود که برنامه‌نویس مایل است به بیش از یک متغیر در آغاز اجرای حلقه تکرار، مقدار اولیه دهد.

در ترتیب اجرای عملگرها حق تقدم عملگر کاما پایین تر از عملگرهای جایگزینی می‌باشد.

ترتیب اجرای عبارت هایی که با کاما جدا شده‌اند، از چپ به راست می‌باشد.

مقدار و نوع لیست عبارت‌هائی که با کاما از هم جدا می‌شوند مقدار و نوع سمت راست‌ترین عبارت داخل لیست است.

مقدمه ۱ while ۲ do...while ۳ for ۴ مقایسه و نکات مهم ۵ قاتمه حلقه ۶ پرش ۷ اعداد شبه تصادفی ۸

## مثال (۱۴) مضور عملگر (.)

```
int main ( )
```

```
{   int i, j, n;
```

```
    cout << "Enter the number n: ";
```

```
    cin >> n;
```

الف: چنانچه  $n$  از ۵ بیشتر باشد،

```
    for ( i = 1, j = 10 ; i < n && j > i ; i++, j-- )
```

```
        cout << i << "t" << j << endl;
```

```
    return 0;
```

```
}
```



1	10
2	9
3	8
4	7
5	6

ب: چنانچه مقدار  $n$  برابر ۵ یا کمتر باشد، فقط  $n-1$  سطر اول از خروجی فوق

چاپ می شود.

31/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016

مقدمه ۱ while ۲ do...while ۳ for ۴ مقایسه و نکات مهم ۵ قاتمه حلقه ۶ پرش ۷ اعداد شبه تصادفی ۸

## مثال (۱۵)

چنانچه  $x$  برابر ۶ و  $y$  برابر ۲ باشد، به هریک از دو عبارت زیر و حاصل آن ها توجه کنید:

$$z = (x-1, y+5) \rightarrow z = (5, 7) \quad z = 7 \rightarrow 7$$

$$z = x-1, y+5 \rightarrow z = 5, 7 \rightarrow 7$$

مقدمه ۱ while ۲ do...while ۳ for ۴ مقایسه و نکات مهم ۵ قاتمه حلقه ۶ پرش ۷ اعداد شبه تصادفی ۸

مثال ۱۶) برنامه‌ای بنویسید که یک جدول ضرب چاپ کند.

```
#include <iomanip> // defines setw()
```

```
#include <iostream> // defines cout
```

```
using namespace std;
```

```
int main()
```

```
{ for (int x=1; x <= 10; x++)
```

```
{ for (int y=1; y <= 10; y++)
```

```
    cout << setw(4) << x*y;
```

```
    cout << endl;
```

```
}
```

```
return 0;
```

```
}
```



۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
۲	۴	۶	۸	۱۰	۱۲	۱۴	۱۶	۱۸	۲۰
۳	۶	۹	۱۲	۱۵	۱۸	۲۱	۲۴	۲۷	۳۰
۴	۸	۱۲	۱۶	۲۰	۲۴	۲۸	۳۲	۳۶	۴۰
۵	۱۰	۱۵	۲۰	۲۵	۳۰	۳۵	۴۰	۴۵	۵۰
۶	۱۲	۱۸	۲۴	۳۰	۳۶	۴۲	۴۸	۵۴	۶۰
۷	۱۴	۲۱	۲۸	۳۵	۴۲	۴۹	۵۶	۶۳	۷۰
۸	۱۶	۲۴	۳۲	۴۰	۴۸	۵۶	۶۴	۷۲	۸۰
۹	۱۸	۲۷	۳۶	۴۵	۵۴	۶۳	۷۲	۸۱	۹۰
۱۰	۲۰	۳۰	۴۰	۵۰	۶۰	۷۰	۸۰	۹۰	۱۰۰

33/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016

مقدمه ۱ while ۲ do...while ۳ for ۴ مقایسه و نکات مهم ۵ قاتمه حلقه ۶ پرش ۷ اعداد شبه تصادفی ۸

قاتمه دادن به یک حلقه

برای پایان دادن به حلقه‌ها قبل از موعده می‌توان از چهار روش اقدام نمود:

break 1

continue 2

exit(0) 3

ctrl + c 4 فشردن کلیدهای



www.RadSms.com

34/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

مثال ۱۷) استفاده از دستور break برای قاتمه دادن به یک حلقه

```
int main()
{
    int n, i=1;
    cout << "Enter a positive integer: ";
    cin >> n;
    long sum=0;
    while (true)
    {
        if (i > n) break;
        sum += i++;
    }
    cout << "The sum of the first " << n << " integers is " << sum << endl;
}
```

یکی از مزیت های دستور break این است  
که فوراً حلقه را قاتمه می دهد بدون این که  
باقی دستورهای درون حلقه اجرا شوند.

Enter a positive integer: 15

The sum of the first 15 integers is 120

35/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

مثال ۱۸) استفاده از تابع exit(0)

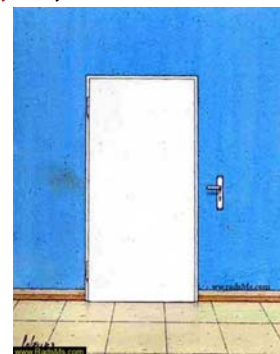
تابع exit(0) روش دیگری برای قاتمه دادن به یک حلقه است. هرچند که این  
تابع بلافاصله اجرای کل برنامه را پایان می دهد؛

```
int main()
{
    long bound;
    cout << "Enter a positive integer: ";
    cin >> bound;
    cout << "Fibonacci numbers < " << bound << ": \n0, 1";
    long f0=0, f1=1;
    while (true)
    {
        long f2 = f0 + f1;
        if (f2 > bound) exit(0);
        cout << ", " << f2;
        f0 = f1;
        f1 = f2;
    }
}
```

Enter a positive integer: 1000

Fibonacci numbers < 1000:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987



36/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016



مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

مثال ۱۹) متوقف کردن یک حلقه نامتناهی

```
int main()
{ long bound;
  cout << "Enter a positive integer: ";
  cin >> bound;
  cout << "Fibonacci numbers < " << bound << ": \n0, 1";
  long f0=0, f1=1;
  while (true)    // Error: Never ends! Press <Ctrl>+c.)
  { long f2 = f0 + f1;
    cout << ", " << f2;
    f0 = f1;
    f1 = f2;
  }
}
```

Enter a positive integer: 125

Fibonacci numbers &lt; 125:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987

159781, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418,

317811, 5040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930

24157817, 63245986, 102334155, 165580141, 267914296, 433494437,

37/47

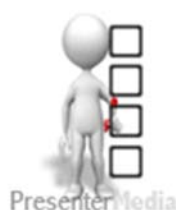
دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

دستور continue

شبه **break** اما به جای اینکه حلقه را قاتمه دهد، اجرا را به تکرار بعدی حلقه منتقل می کند. این دستور، ادامه پرفه فعلی را لغو کرده و اجرای دور بعدی حلقه را آغاز می کند.



38/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

## مثال ۲۰) استفاده از دستور continue

```
int main()
{
    int n = 1;
    char c;
    for( ; ; n++ )
    {
        cout << "\nLoop no: " << n << endl;
        cout << "Continue? <y|n> ";
        cin >> c;
        if (c == 'y') continue;
        break;
    }
    cout << "\nTotal of loops: " << n;
}
```

Loop no: 1  
Continue? y  
Loop no: 2  
Continue? y  
Loop no: 3  
Continue? n  
Total of loops: 3

39/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

## دستورهای پرش، goto

دستورهای break ، switch و continue: انتقال اجرای برنامه به مکانی غیر از

جایی که به طور طبیعی باید می رفت

مقصد انتقال: نوع دستور

break: به خارج از حلقه می رود،

continue: به شرط ادامه حلقه (دور بعدی حلقه) می رود،

switch: به یکی از ثابت های case می رود،

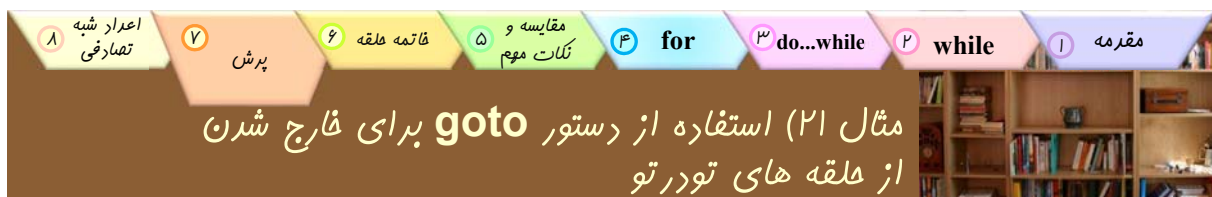
goto: به یک برچسب معین می رود.



**برپرسب** شناسه‌ای است که جلوی آن **علامت کولن (:)** می‌آید و جلوی یک دستور دیگر قرار می‌گیرد و مقصد پرش را مشخص می‌کند.

**مزیت (goto):** با استفاده از آن می‌توان از همه حلقه‌های تودرتو خارج شد و به مکان دلخواهی در برنامه پرش نمود.

**عیب:** تجربه نشان داده که استفاده غیر محتاطانه از دستور goto سبب افزایش خطاهای زمان اجرا و کاهش پایداری برنامه می‌شود.



مثال (۲۱) استفاده از دستور goto برای خارج شدن از حلقه های تودرتو

```
int main()
```

```
{ const int N=5;
```

```
for (int i=0; i<N; i++)
```

```
{ for (int j=0; j<N; j++)
```

```
{ for (int k=0; k<N; k++)
```

```
if (i+j+k>N) goto esc;
```

```
else cout << i+j+k << " ";
```

```
cout << "*" << " ";
```

```
}
```

```
esc: cout << "." << endl; // inside the i loop, outside the j loop
```

```
}
```

```
}
```

```
0 1 2 3 4 * 1 2 3 4 5 * 2 3 4 5 .
```

```
1 2 3 4 5 * 2 3 4 5 .
```

```
2 3 4 5 .
```

```
3 4 5 .
```

```
4 5 .
```

دستور break فقط درونی‌ترین حلقه را قاتمه می‌دهد ولی با دستور goto می‌توان چند حلقه یا همه حلقه‌ها را یک جا قاتمه داد.

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

## تولید اعداد شبه تصادفی

در شبیه سازی نیاز است «اعداد تصادفی» توسط رایانه ها تولید شود تا نادرانسته های دنیای واقعی مدل سازی شود. البته رایانه ها «ثابت کار» هستند. یعنی با دادن داده های مشابه به رایانه های مشابه، همیشه خروجی یکسان تولید می شود.

«اعداد شبه تصادفی»: ← اعدادی به ظاهر تصادفی که به طور یکنواخت در یک محدوده خاص گسترده اند.

سرفایل <cstdlib>؛ تابع rand() تابع اعداد صحیح شبه تصادفی در محدوده صفر تا RAND\_MAX.

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

## مثال ۲۲) برنامه ای بنویسید که اعداد شبه تصادفی تولید کند.

```
#include <cstdlib> // defines the rand() and RAND_MAX
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{ // prints pseudo-random numbers:
```

```
for (int i = 0; i < 8; i++)
```

```
cout << rand() << endl;
```

```
cout << "RAND_MAX = " << RAND_MAX << endl;
```

```
}
```

```
1103527590
```

```
377401575
```

```
662824084
```

```
1147902781
```

```
2035015474
```

```
368800899
```

```
1508029952
```

```
486256185
```

```
RAND_MAX = 2147483647
```

```
1103527590
```

```
377401575
```

```
662824084
```

```
1147902781
```

```
2035015474
```

```
368800899
```

```
1508029952
```

```
486256185
```

```
RAND_MAX = 2147483647
```



PresenterMedia

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

مثال ۲۳) هسته به طور مفاوره ای

```
#include <cstdlib> // defines the rand() and srand()
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{ // prints pseudo-random numbers:
```

```
    unsigned seed;
```

```
    cout << "Enter seed: ";
```

```
    cin >> seed;
```

```
    srand(seed);
```

```
    // initializes the seed
```

```
    for (int i = 0; i < 8; i++)
```

```
        cout << rand() << endl;
```

```
}
```

سه اجرای متفاوت از برنامه بالا نشان

داره شده است:

Enter seed: 0

12345

1406932606

654583775

1449466924

229283573

1109335178

1051550459

1293799192

Enter seed: 1

1103527590

377401575

662824084

1147902781

2035015474

368800899

1508029952

486256185

Enter seed: 12345

1406932606

654583775

1449466924

229283573

1109335178

1051550459

1293799192

794471793

45/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016

مقدمه ۱ while ۲ do...while ۳ for ۴ نکات مهم ۵ مقایسه و ۶ قاتمه حلقه ۷ پرش ۸ اعداد شبه تصادفی

مثال ۲۴) استفاده از ساعت سیستم برای هسته

```
#include <cstdlib> // defines the rand() and srand()
```

```
#include <ctime> // defines the time() function
```

```
#include <iostream>
```

```
using namespace std;
```

```
//#include <time.h> // use this if <ctime> is not recognized
```

```
int main()
```

```
{ // prints pseudo-random numbers:
```

```
    unsigned seed = time(NULL); // uses the system clock
```

```
    cout << "seed = " << seed << endl;
```

```
    srand(seed); // initializes the seed
```

```
    for (int i = 0; i < ۷; i++)
```

```
        cout << rand() << endl;
```

```
}
```

seed = 808148157

1877361330

352899587

1443923328

1857423289

۲۰۰۳۹۸۸۴۶

۱۳۷۹۶۹۹۵۵۱

۱۶۲۲۷۰۲۵۰۸

seed = 808148160

892939769

1559273790

1468644255

952730860

1322627253

844657339

440402904

46/47

دانشگاه صنعتی امیرکبیر - دکتر فرناز قاسمی

13/11/2016





پرسش و پاسخ ؟