

حلقه‌ی تکرار ۲

مبانی برنامه‌نویسی

(۱۱-۱۳-۱۳۹)

جلسه‌ی پانزدهم



دانشگاه شهید بهشتی

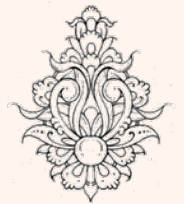
پاییز ۱۳۹۳

دانشکده‌ی مهندسی برق و کامپیوتر

احمد محمودی ازناوه

فهرست مطالب

- حلقه‌ها
- حلقه‌های تو در تو



اعداد فیبوناچی

- اعداد فیبوناچی به اعدادی گویند که از رابطه‌ی زیر پیروی نمایند:

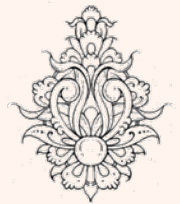
$$\begin{cases} F_0 = 0 \\ F_1 = 1 \\ F_n = F_{n-1} + F_{n-2} \end{cases}$$

- در این حالت برای $n=2,3,4$ خواهیم داشت:

$$F_2 = F_{2-1} + F_{2-2} = F_1 + F_0 = 0 + 1 = 1$$

$$F_3 = F_{3-1} + F_{3-2} = F_2 + F_1 = 1 + 1 = 2$$

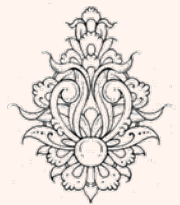
$$F_4 = F_{4-1} + F_{4-2} = F_3 + F_2 = 2 + 1 = 3$$



ملقهى بى پايان

```
#include <iostream>
using namespace std;
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;
    cout << ", " << f2;
    f0 = f1;
    f1 = f2;
}
}
```

```
Enter a positive integer: 4
Fibonacci numbers < 4:
0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987,1597,2584,4181,6765,10946,17711,
,28657,46368,75025,121393,196418,317811,514229,832040,1346269,2178309,3524578,57
02887,9227465,14930352,24157817,39088169,63245986,102334155,165580141,267914296,
433494437,701408733,1134903170,1836311903,-1323752223,512559680,-811192543,-2986
```



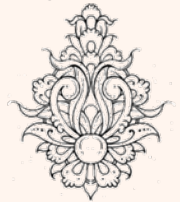
اعداد فیبوناچی (ادامه...)

– برنامه‌ای بنویسید که عددی را به عنوان مقدار سرحد دریافت کرده سری را تا زمانی که عضو سری از مقدار سرحد کمتر است محاسبه و چاپ نماید.

```
#include <iostream>
using namespace std;
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)
            break; // terminates the loop immediately
        cout << ", " << f2;
        f0 = f1;
        f1 = f2;
    }
}
```

```
Enter a positive integer: 1234
Fibonacci numbers < 1234:
0,1,1,2,3,5,8,13,21,34,55,89,144,233,377,610,987
```

| n | F_n |
|-----|-------|
| 0 | 0 |
| 1 | 1 |
| 2 | 1 |
| 3 | 2 |
| 4 | 3 |
| 5 | 5 |
| 6 | 8 |
| 7 | 13 |
| 8 | 21 |
| 9 | 35 |

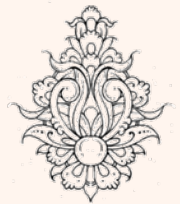


تابع exit

- تابع exit سبب می‌شود برنامه پایان یابد.

```
#include <iostream>
#include <cstdlib>
using namespace std;
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); // terminates the program immediately
        cout << ", " << f2;
        f0 = f1;
        f1 = f2;
    }
}
```

```
Enter a positive integer: 167
Fibonacci numbers < 167:
0,1,1,2,3,5,8,13,21,34,55,89,144
```



یکی از کاربردهای do-while، استفاده از آن در تولید منوهای ساده در برنامه است.

```
int main()
{
int choice, months;
double charges;
do
{ // Display the menu choices
cout << "\n\tHealth Club Membership Menu\n\n";
cout << "1. Standard Adult Membership\n";
cout << "2. Child Membership\n";
cout << "3. Senior Citizen Membership\n";
cout << "4. Quit the Program\n\n";
cout << "Enter your choice: ";
cin >> choice;
if (choice >= 1 && choice <= 3)
{ cout << "For how many months? ";
cin >> months;
switch (choice)
{ case 1: charges = months * 40.0;
break;
case 2: charges = months * 20.0;
break;
case 3: charges = months * 30.0;
}
cout << "The total charges are $" << charges << endl;
}
else if (choice != 4){
cout << "The valid choices are 1 through 4.\n";
cout << "Run the program again, select one of these
}
} while (choice != 4);
return 0;
}
```

```
Health Club Membership Menu
1. Standard Adult Membership
2. Child Membership
3. Senior Citizen Membership
4. Quit the Program
Enter your choice: 1
For how many months? 13
The total charges are $520

Health Club Membership Menu
1. Standard Adult Membership
2. Child Membership
3. Senior Citizen Membership
4. Quit the Program
Enter your choice: 3
For how many months? 4
The total charges are $120

Health Club Membership Menu
1. Standard Adult Membership
2. Child Membership
3. Senior Citizen Membership
4. Quit the Program
Enter your choice: 5
The valid choices are 1 through 4.
Run the program again, select one of these.

Health Club Membership Menu
1. Standard Adult Membership
2. Child Membership
3. Senior Citizen Membership
4. Quit the Program
Enter your choice: 4
Press any key to continue . . . _
```

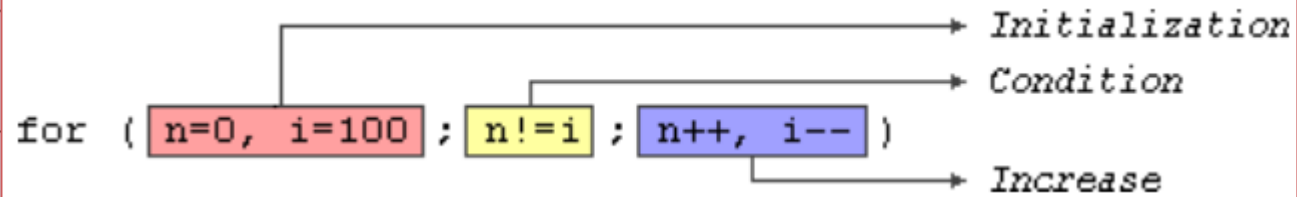
expression1

As long as expression2 is true

statement

expression3

حلقه‌ی for



```
for( expression1, expression2, expression3 )
    statement
```

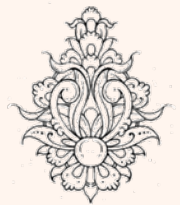
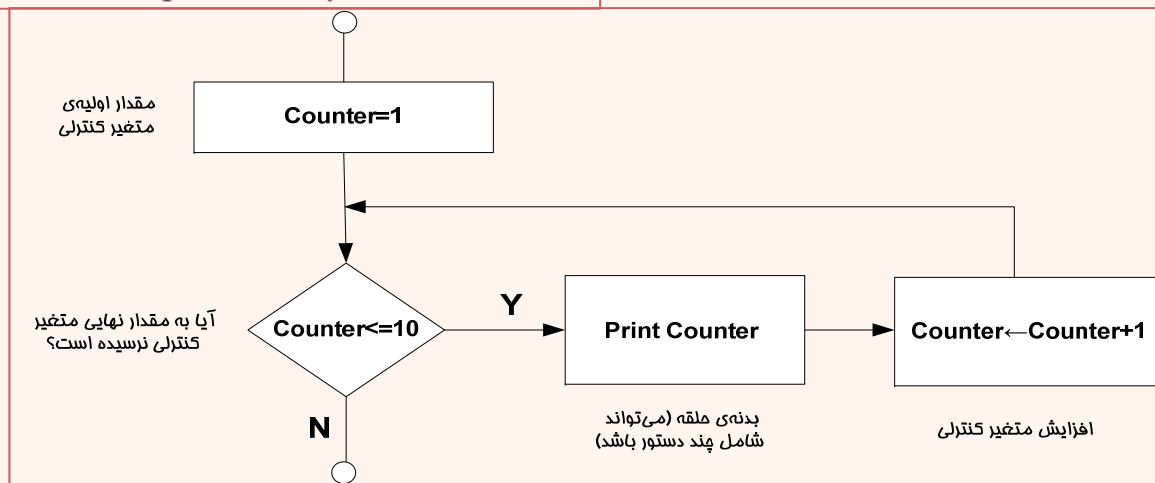
Step 1. Perform the initialization expression.

Step 2. Evaluate the test expression.
If it is true, go to step 3.
Otherwise, terminate the loop.

Step 4. Perform the update expression. Go back to step 2.

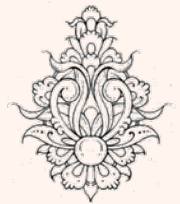
```
for (num = 1; num <= 10; num++)
    cout << num << "\t\t" << (num * num) << endl;
```

Step 3. Execute the body of the loop.




```
#include <iostream>
using namespace std;
int main ()
{
    for (int n=10; n>0; n--) {
        cout << n << ", ";
    }
    cout << "FIRE!\n";
    return 0;
}
```

10, 9, 8, 7, 6, 5, 4, 3, 2, 1, FIRE!



حلقه‌ی for (ادامه...)

مثال

```
#include <iostream>
using namespace std;
int main()
{
double rate = 1.15; // Exchange rate:
// one Euro to one Dollar
cout << "\tEuro \tDollar\n";
for( int euro = 1; euro <= 5; ++euro)
    cout << "\t " << euro << "\t " << euro*rate << endl;
return 0;
}
```

| Euro | Dollar |
|------|--------|
| 1 | 1.15 |
| 2 | 2.3 |
| 3 | 3.45 |
| 4 | 4.6 |
| 5 | 5.75 |



مقداردهی و مقداردهی مجدد

```
#include <iostream>
using namespace std;
int main()
{
    int count = 1;
    while( count <= 10)
    {
        cout << count << ". loop" << endl;
        ++count;
    }
}
```

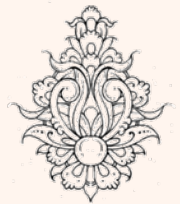
```
1. loop
2. loop
3. loop
4. loop
5. loop
6. loop
7. loop
8. loop
9. loop
10. loop
```



مقداردهی و مقداردهی مجدد (ادامه...)

```
#include <iostream>
using namespace std;
int main()
{
int count;
for( count = 1; count <= 10; ++count)
    cout << count << ". loop" << endl;
}
```

```
1. loop
2. loop
3. loop
4. loop
5. loop
6. loop
7. loop
8. loop
9. loop
10. loop
```



مثال

```
1. loop
2. loop
3. loop
4. loop
5. loop
6. loop
7. loop
8. loop
9. loop
10. loop
```

```
#include <iostream>
using namespace std;
int main()
{
int count;
for( count = 1; count <= 10; cout<<count++ <<".loop" << endl)
;
}
```

```
1. loop
1--2. loop
2--3. loop
3--4. loop
4--5. loop
5--6. loop
6--7. loop
7--8. loop
8--9. loop
9--10. loop
10--ahmad@ubunt
```

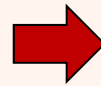
```
#include <iostream>
using namespace std;
int main()
{
int count;
for( count = 1; count <= 10; cout << count++ <<"--"
cout << count << ". loop" << endl;
}
```

```
for(;;)
```

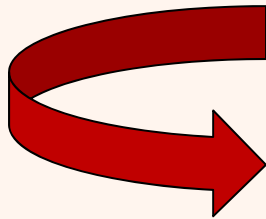


مجادل ملقهی بی‌نهایت است هر
کدام از عبارات در ملقهی *for*
می‌تواند حذف گردد.

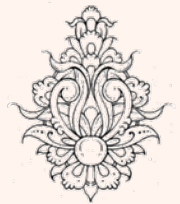
```
for( ; expression; )
```



مجادل ملقهی است که تا وقتی
که شرط برقرار باشد ادامه
می‌یابد.



مجادل ملقهی
While(expression)



از کدام حلقه استفاده کنیم؟

- اگر چه می‌توان تکرار را در الگوریتم‌های مختلف با هر یک از حلقه‌های ذکر شده پیاده‌سازی نمود، استفاده از هر یک در موقعیت‌هایی خاص مناسب‌تر است.

`while` loop is a *pretest* loop

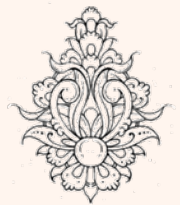
• حلقه‌ی `while`

– حلقه‌ی `while` یک حلقه‌ی «پیش‌آزمونی» است. برای مواردی استفاده می‌شود که اگر شرط در ابتدا هم نادرست باشد اجرا نگردد.

– در مواردی که نیاز به «نگهبان» است نیز کاربرد دارد.

sentinel

A *sentinel* is a special value that marks the end of a list of values.



مثال while

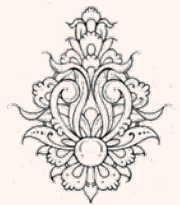
```
int main()
{
    int game = 1,
    points,
    total = 0;

    cout << "Enter the number of points your team has earned\n";
    cout << "so far in the season, then enter -1 when finished.\n\n";
    cout << "Enter the points for game " << game << ": ";
    cin >> points;
    while (points != -1)
    {
        total += points;
        cout << "Enter the points for game " << ++game << ": ";
        cin >> points;
    }
    cout << "\nThe total points are " << total << endl;
    return 0;
}
```

```
Enter the number of points your team has earned
so far in the season, then enter -1 when finished.
```

```
Enter the points for game 1: 2
Enter the points for game 2: 33
Enter the points for game 3: 4
Enter the points for game 4: 5
Enter the points for game 5: -1
```

```
The total points are 44
```



do-while loop is a *posttest* loop.

Do-while •

– حلقه‌ی do-while حلقه‌ی است «پس‌آزمونی»، در جایی مناسب است که دست کم یک‌بار انتظار داریم حلقه اجرا شود.

```
char doAgain;
int num;
cout << "This program finds the square of any integer.\n";
do{
    cout << "\nEnter an integer: ";
    cin >> num;
    cout << num << " squared is " << pow(num, 2) << endl;
    cout << "Do you want to square another number? (Y/N) ";
    cin >> doAgain;
} while (doAgain == 'Y' || doAgain == 'y');
```



حلقه‌ی for

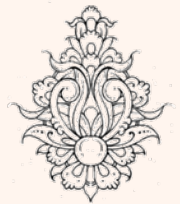
for loop is a *pretest* loop that first executes an initialization expression.

• حلقه‌ی for

– یک حلقه‌ی پیش‌آزمونی است که ابتدا بر اساس مقدار دهی اولیه اجرا می‌شود

– برای زمانی که تعداد تکرار مشخص است کاربرد دارد

```
cout << "This program finds the squares of the integers "  
    << "from 1 to 8.\n\n";  
for (num = 1; num <= 8; num++)  
{  
    cout << num << " squared is " << pow(num, 2) << endl;  
}
```

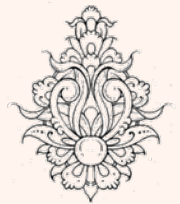


متوقف‌سازی حلقه

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

int main ()
{
    int n;
    for (n=10; n>0; n--)
    {
        cout << n << ", ";
        if (n==3)
        {
            cout << "countdown aborted!";
            break;
        }
    }
    return 0;
}
```

10, 9, 8, 7, 6, 5, 4, 3, countdown aborted!

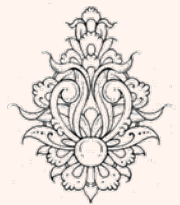


دستور continue

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

int main ()
{
    for (int n=10; n>0; n--) {
        if (n==5) continue;
        cout << n << ", ";
    }
    cout << "FIRE!\n";
    return 0;
}
```

10, 9, 8, 7, 6, 4, 3, 2, 1, FIRE!

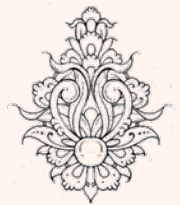
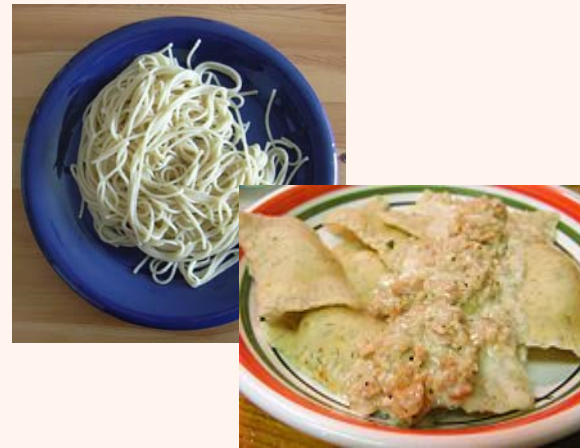


دستور go-to

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

int main ()
{
    int n=10;
loop:
    cout << n << ", ";
    n--;
    if (n>0) goto loop;
    cout << "FIRE!\n";
    return 0;
}
```

- در برنامه نویسی ساخت یافته استفاده از این دستور درست نیست. برنامه هایی که از این دستور استفاده می کنند به Spaghetti code موسومند.



10, 9, 8, 7, 6, 5, 4, 3, 2, 1, FIRE!