

ساختارهای تصمیم‌گیری

طاقه‌ها

مبانی برنامه‌نویسی

(۱۱-۱۳۹-۱۳۹۹)

جلسه‌ی شانزدهم



دانشگاه شهید بهشتی

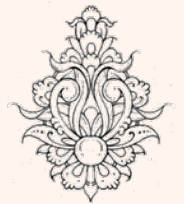
پاییز ۱۳۹۲

دانشکده‌ی مهندسی برق و کامپیوتر

احمد محمودی ازناوه

# فهرست مطالب

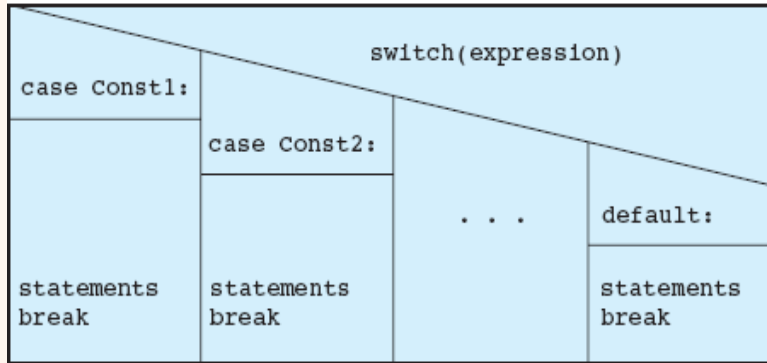
- switch
- حلقه‌ها
- خروج از حلقه
- آشنایی با کتابخانه‌ی math.h



# انتخاب با دستور Switch

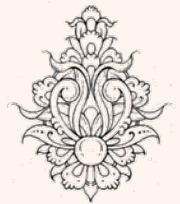
– به جای دستور if else استفاده می‌گردد.

– ساختار:

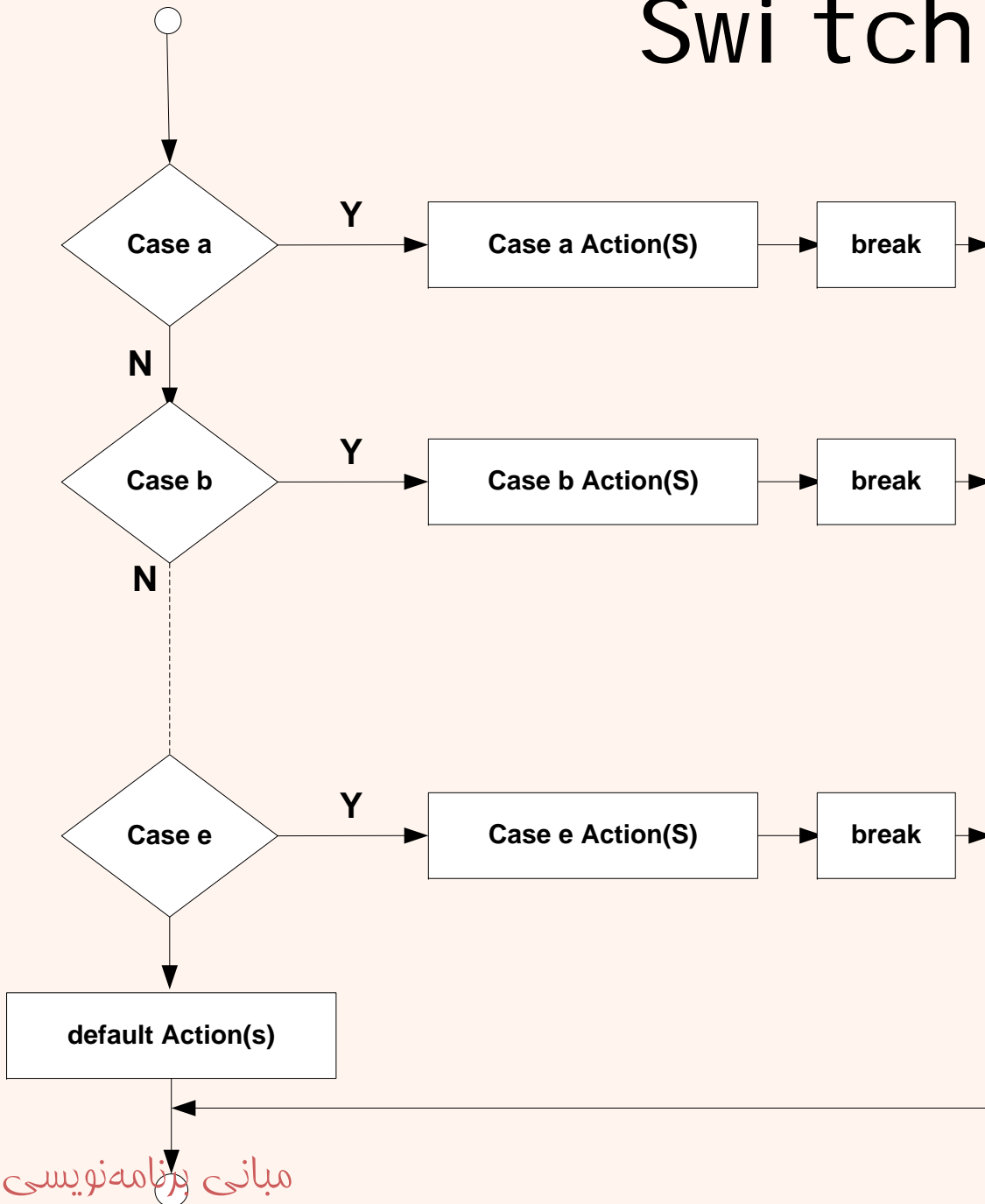


**Switch ( expression )**

```
{  
    case const1: [ statement ; ]  
                [ break ; ]  
    case const2: [ statement ; ]  
                [ break ; ]  
    .  
    .  
    .  
    [ default : statement ]  
}
```



# انتخاب با دستور Switch



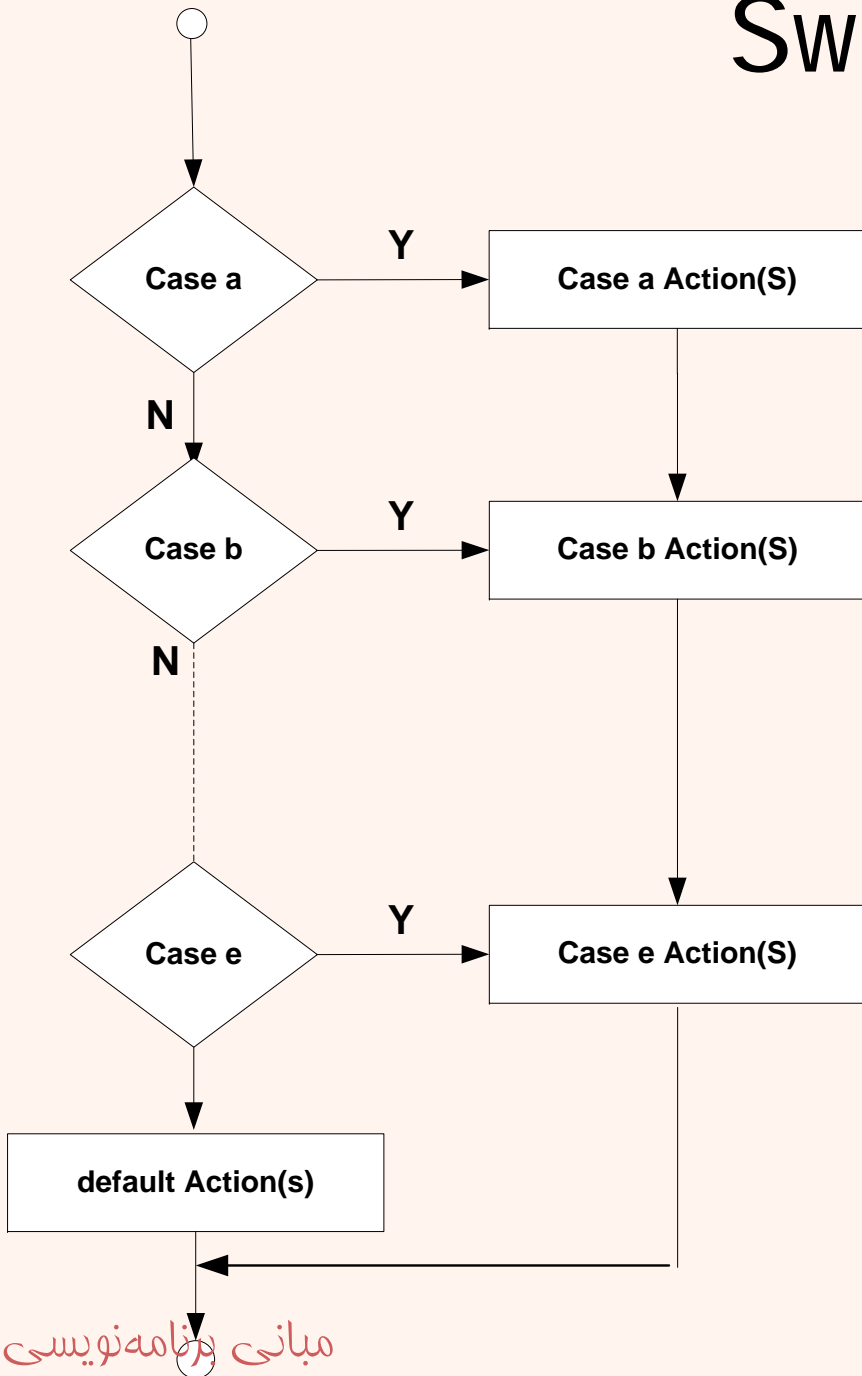
استفاده از این دستور  
خوانایی کد را افزایش  
می دهد

کد راحت تر فهمیده می شود

باز نویسی و اعمال اصلاحات  
ساده تر خواهد بود

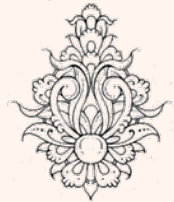


# انتخاب با دستور Switch



در صورتی که از *break* استفاده نکنیم، خروجی آن مطابق شکل رویی خواهد بود

**Fall through**



## مثال switch

```
switch (x) {  
case 1:  
    cout << "x is 1";  
break;  
case 2:  
    cout << "x is 2";  
break;  
default:  
    cout << "value of x  
unknown";  
}
```

## معادل if-else

```
if (x == 1) {  
    cout << "x is 1";  
}  
else if (x == 2) {  
    cout << "x is 2";  
}  
else{  
    cout << "value of x  
unknown";  
}
```



```

int main()
{ int score;
cout << "Enter your test score: "; cin >> score;
switch (score/10)
{ case 10:
case 9: cout << "Your grade is an A." << endl; // LOGICAL ERROR
case 8: cout << "Your grade is a B." << endl; // LOGICAL ERROR
case 7: cout << "Your grade is a C." << endl; // LOGICAL ERROR
case 6: cout << "Your grade is a D." << endl; // LOGICAL ERROR
case 5:
case 4:
case 3:
case 2:
case 1:
case 0: cout << "Your grade is an F." << endl; // LOGICAL ERROR
default: cout << "Error: score is out of range.\n";
}
cout << "Goodbye." << endl;
}

```



```

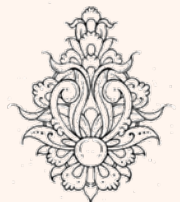
Enter your test score: 83
Your grade is a B.
Your grade is a C.
Your grade is a D.
Your grade is an F.
Error: score is out of range.
Goodbye.

```



```
int main()
{ int score;
cout << "Enter your test score: "; cin >> score;
switch (score/10)
{ case 10:
case 9: cout << "Your grade is an A." << endl; break;
case 8: cout << "Your grade is a B." << endl; break;
case 7: cout << "Your grade is a C." << endl; break;
case 6: cout << "Your grade is a D." << endl; break;
case 5:
case 4:
case 3:
case 2:
case 1:
case 0: cout << "Your grade is an F." << endl; break;
default: cout << "Error: score is out of range.\n";
}
cout << "Goodbye." << endl;
}
```

```
Enter your test score: 83
Your grade is a B.
Goodbye.
```



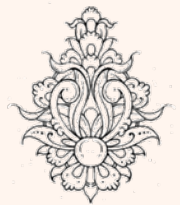


# تمرین

- حاصل این کد با ورودی ۱۴ چه خواهد بود؟

```
int main()
{ int score;
cout << "Enter your test score: "; cin >> score;
switch (score/10)
{ case 10:
case 9: cout << "Your grade is an A." << endl; break;
case 8: cout << "Your grade is a B." << endl; break;
case 7: cout << "Your grade is a C." << endl; break;
case 6: cout << "Your grade is a D." << endl; break;
case 5:
case 4:
case 3:
case 2:
case 1:
case 0: cout << "Your grade is an F." << endl; break;
default: cout << "Error: score is out of range.\n";
}
cout << "Goodbye." << endl;
}
```

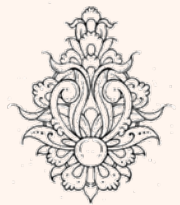
```
Enter your test score:44
Your grade is an F.
Goodbye.
```



## تمرین

- برنامه‌ای بنویسید که دو عدد صحیح را از ورودی بگیرد سپس عملگر مناسب را نیز گرفته عملیات متناظر را صورت دهد.

```
Enter two integers: 3 5
Enter an operator: +
8
```



```

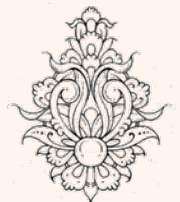
#include <iostream>
using namespace std;
int main(){
    int x,y;
    char op;
    cout << "Enter two Numbers: ";
    cin >> x >> y;
    cout << "Enter an operator: ";
    cin >> op;
    switch (op){
        case '+': cout << x + y << endl; break;
        case '-': cout << x - y << endl; break;
        case '*': cout << x * y << endl; break;
        case '/': cout << x / y << endl; break;
        case '%': cout << x % y << endl;
    }
}

```

```

Enter two integers: 3 5
Enter an operator: +
8

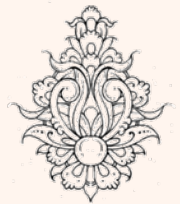
```



ادامه...

```
#include <iostream>
using namespace std;
int main(){
    int x,y;
    char op;
    cout << "Enter two Numbers: ";
    cin >> x >> y;
    cout << "Enter an operator: ";
    cin >> op;
    switch (op){
        case '+': cout << x + y << endl; break;
        case '-': cout << x - y << endl; break;
        case '*': cout << x * y << endl; break;
        case '/': cout << x / y << endl; break;
        case '%': cout << x % y << endl;
    }
}
```

• سوال



– در صورت تعریف نوع اعشاری برای دو عدد ورودی حاصل کدام فرآیندها اگر ورودی ۵ و ۳ باشد متفاوت است؟



```
Enter two integers: 5 3
Enter an operator: /
```

صمیم

```
Enter two Numbers: 5 3
Enter an operator: /
1.66667
```

اعشاری

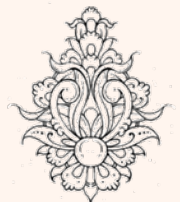
ادامه...

• نکته:

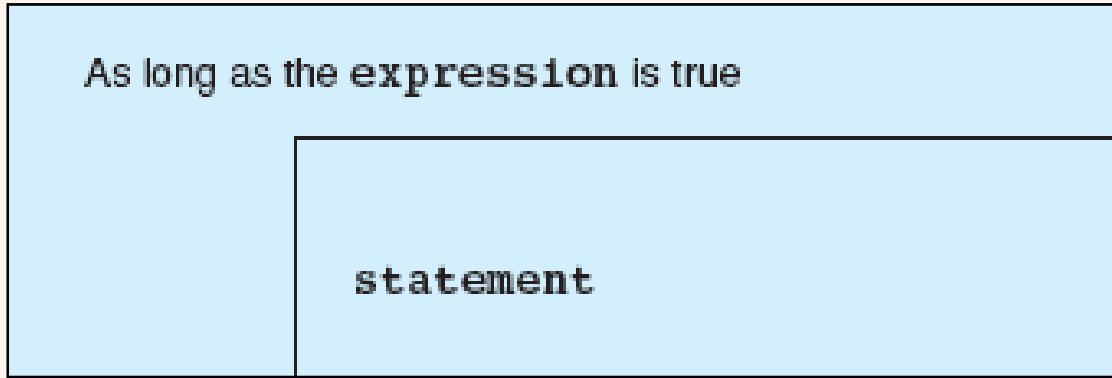
– عملگر % برای اعداد اعشاری با خطا مواجه می‌شود

```
#include <iostream>
using namespace std;
int main() {
    float x,y;
    char op;
    cout << "Enter two Numbers: ";
    cin >> x >> y;
    cout << "Enter an operator: ";
    cin >> op;
    switch (op){
        case '+': cout << x + y << endl; break;
        case '-': cout << x - y << endl; break;
        case '*': cout << x * y << endl; break;
        case '/': cout << x / y << endl; break;
        case '%': cout << x % y << endl;
    }
}
```

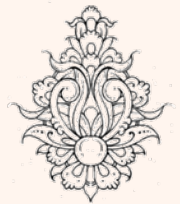
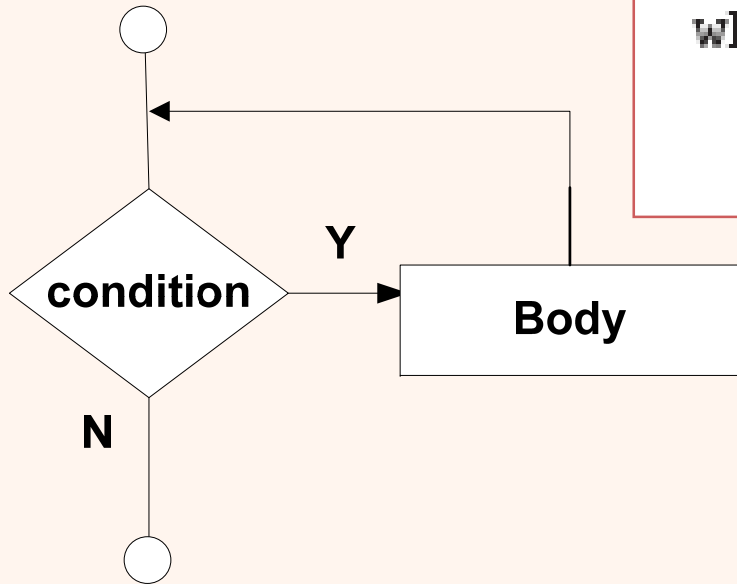
```
error C2296: '%' : illegal, left operand has type 'float'
error C2297: '%' : illegal, right operand has type 'float'
```



# حلقه‌ی While



```
while( expression )  
    statement // loop body
```



اگر در حلقه بیش از یک دستور باشد، می‌باید از بلوک استفاده شود.

# مثال

برنامه‌ای بنویسید که عدد صحیح و مثبت  $n$  را از ورودی بخواند و اعداد  $n$  تا ۱ را به صورت نزولی چاپ کند

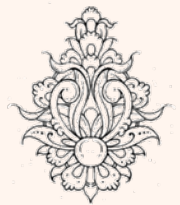
```
#include <iostream>
using namespace std;

int main ()
{
    int n;
    cout << "Enter the starting number > ";
    cin >> n;

    while (n>0) {
        cout << n << ", ";
        --n;
    }

    cout << "FIRE!\n";
    return 0;
}
```

```
Enter the starting number > 10
10, 9, 8, 7, 6, 5, 4, 3, 2, 1, FIRE!
```



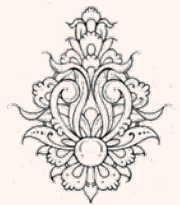
# مثال

برنامه‌ای بنویسید که عدد صحیح و مثبت  $n$  را از ورودی بخواند و بدون استفاده از فرمول تصاعد حسابی، مجموع اعداد آن  $n$  را محاسبه و چاپ 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;
}
```

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

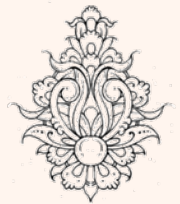
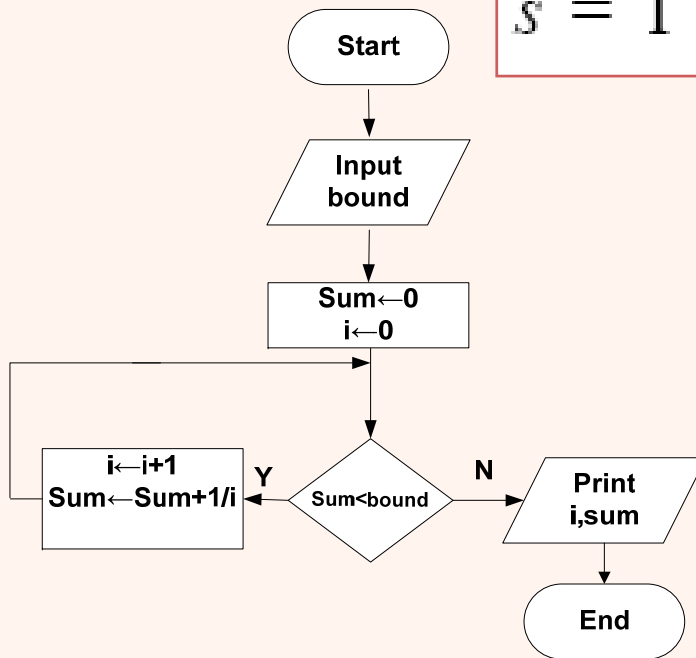
```
Enter a positive integer: 8
The sum of the first 8 integers is 36
```





عددی را به عنوان حد (bound) وارد کرده حاصل عبارت زیر را تا زمانی که حاصل جمع از bound بیشتر یا مساوی شود محاسبه کنید. در آخر مجموع و تعداد جملات سری را چاپ کنید.

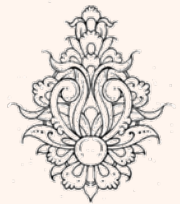
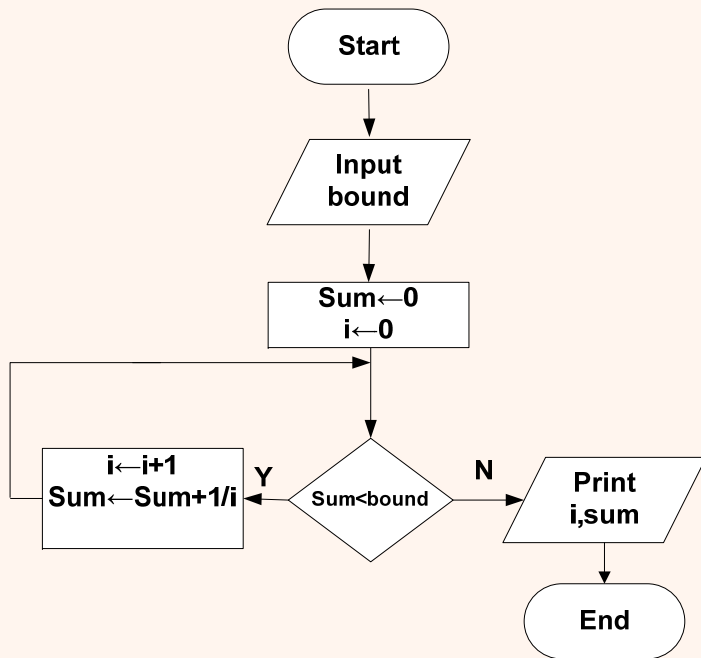
$$s = 1 + 1/2 + 1/3 + \dots + 1/n$$



# مثال (ادامه...)

```
#include <iostream>
using namespace std;
int main(){
    float bound;
    cout << "Enter a positive number: ";
    cin >> bound;
    double sum=0;
    int i=0;
    while (sum < bound)
        sum += 1.0/++i;
    cout << "The sum of the first " << I
    << " reciprocals is " << sum << endl;
}
```

i	sum
0	0.00000
1	1.00000
2	1.50000
3	1.83333
4	2.08333
5	2.28333
6	2.45000
7	2.59286
8	2.71786
9	2.82897
10	2.92897
11	3.01988



Enter a positive integer: 3  
The sum of the first 11 reciprocals is 3.01988

# مثالی دیگر از حلقه `while`

برنامه‌های بنویسید که هر بار عددی از ورودی بخواند و جذر آن را نمایش دهد، (وارد کردن عددی کوچک‌تر مابقی صفر به معنای پایان کار است).

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

```
Enter a positive number: 49
sqrt(49) = 7
Enter another positive number (or 0 to quit): 9
sqrt(9) = 3
Enter another positive number (or 0 to quit): 81
sqrt(81) = 9
Enter another positive number (or 0 to quit): 34
sqrt(34) = 5.83095
Enter another positive number (or 0 to quit): 0
```



# cmath (math.h)

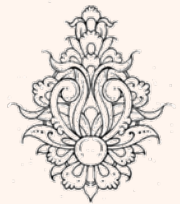
• این کتابخانه شامل تعدادی از توابع پایه می‌باشد:

- fabs(x), abs(n)
- sqrt(x), pow(x,y)
- ceil(x), floor(x)
- exp(x), log(x), log10(x)
- sin(x), cos(x), tan(x)
- asin(x), acos(x), atan(x), atan2(y,x)

تمرین: رابطه‌ی زیر را با استفاده از توابع کتابخانه‌ی cmath بنویسید.

$$Velocity = \sqrt{v_0^2 + 2a(x - x_0)}$$

```
velocity = sqrt(pow(v0,2) + 2*a*(x - x0));
```



# do-while

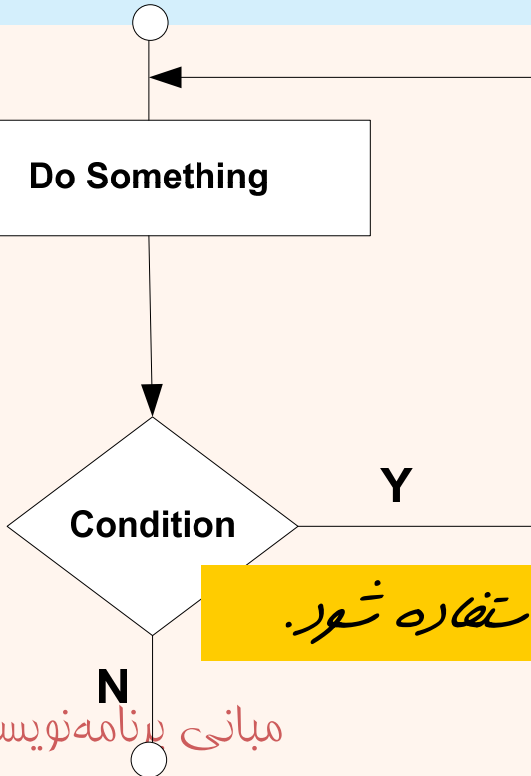
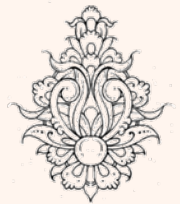
برنامه‌های بنویسید که هر بار عددی از ورودی بخواند، آن را چاپ کند، سپس در صورتی که آن عدد مخالف صفر بود، این روند را ادامه دهد.

statement

حلقه‌ی do-while تنها می‌باید با ; همراه باشد

As long as the expression is true

```
do  
    statement  
while ( expression ) ;
```



اگر در حلقه بیش از یک دستور باشد، می‌باید از بلوک استفاده شود.

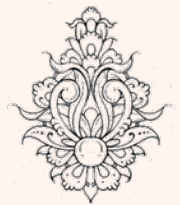
# do-while (مثال)

برنامه‌های بنویسید که هر بار عددی از ورودی بخواند، آن را چاپ کند، سپس در صورتی که آن عدد مخالف صفر بود، این روند را ادامه دهد.

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

int main ()
{
    unsigned long n;
    do {
        cout << "Enter number (0 to end): ";
        cin >> n;
        cout << "You entered: " << n << "\n";
    } while (n != 0);
    return 0;
}
```

```
Enter number (0 to end): 23
You entered: 23
Enter number (0 to end): 3
You entered: 3
Enter number (0 to end): 2
You entered: 2
Enter number (0 to end): 0
You entered: 0
```

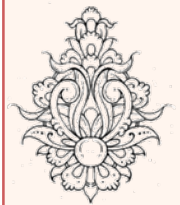


# do-while

برنامه‌های بنویسید که عدد صحیح و مثبت  $n$  را از ورودی بخواند و بدون استفاده از حرمول تصاعد حسابی، مجموع اعداد آن  $n$  را محاسبه و چاپ کند (این بار با استفاده از do-while).

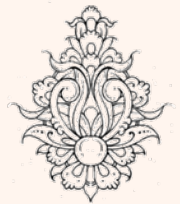
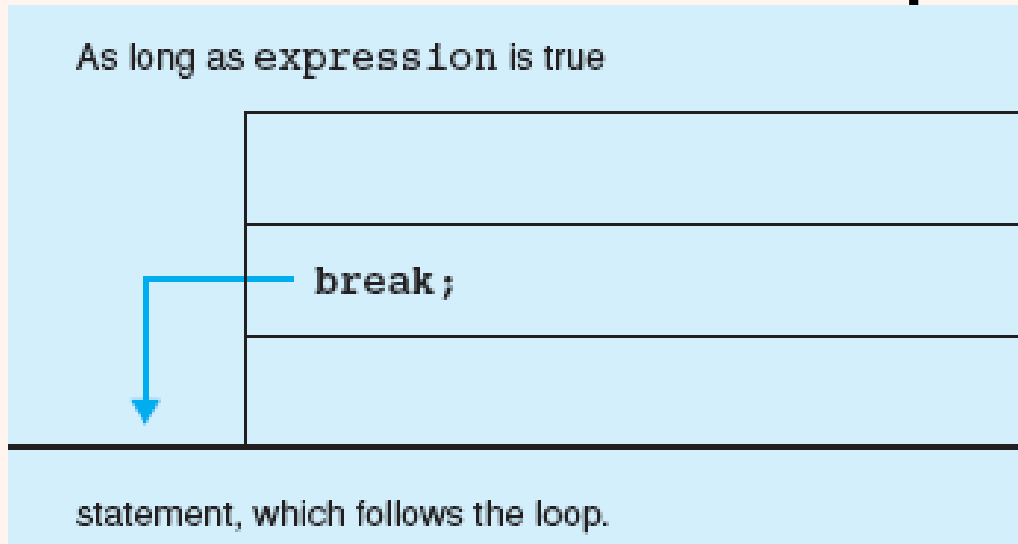
```
int main()
{ int n,i=0;
cout << "Enter a positive integer: ";
cin >> n;
long sum=0;
do{
    sum += i++;
}
while (i <= n);
cout << "The sum of the first " << n << " integers is " << sum << endl;
}
```

```
Enter a positive integer: 8
The sum of the first 8 integers is 36
```



## فاتمه دادن به حلقه

- در دستور switch با break آشنا شدیم،
- دستور break باعث می‌شود از داخلی‌ترین حلقه خارج شویم.
- با اجرای این دستور برنامه به اجرای دستور بعد از حلقه خواهد پرداخت.



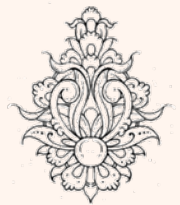


## • پیاده‌سازی مثال جمع n عدد نخست:

```
int main()
{ int n,i=1;
cout << "Enter a positive integer: ";
cin >> n;
long sum=0;
while (true)
{
    if (i > n) break; // terminates the loop immediately
    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
```

```
Enter a positive integer: 100
The sum of the first 100 integers is 5050
```



# اعداد فیبوناچی

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

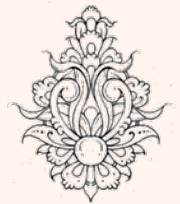
$$\begin{cases} F_0 = 0 \\ F_1 = 1 \\ F_n = F_{n-1} + F_{n-2} \end{cases}$$

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

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

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

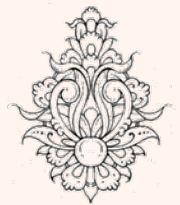
$$F_4 = F_{4-1} + F_{4-2} = F_3 + F_2 = 2 + 1 = 3$$



# 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;
```

```
    }
```

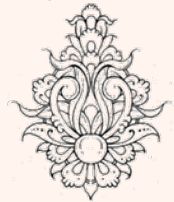
```
}
```

```
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>
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);
        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
```

