

مبانی برنامه‌نویسی

(۱۱-۱۳-۱۳۹۱)

جلسه‌ی چهاردهم

روزنامه‌ها



دانشگاه شهید بهشتی

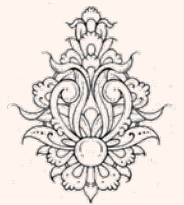
پاییز ۱۳۹۱

دانشکده‌ی مهندسی برق و کامپیوتر

احمد محمودی ازناوه

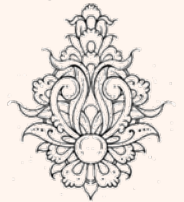
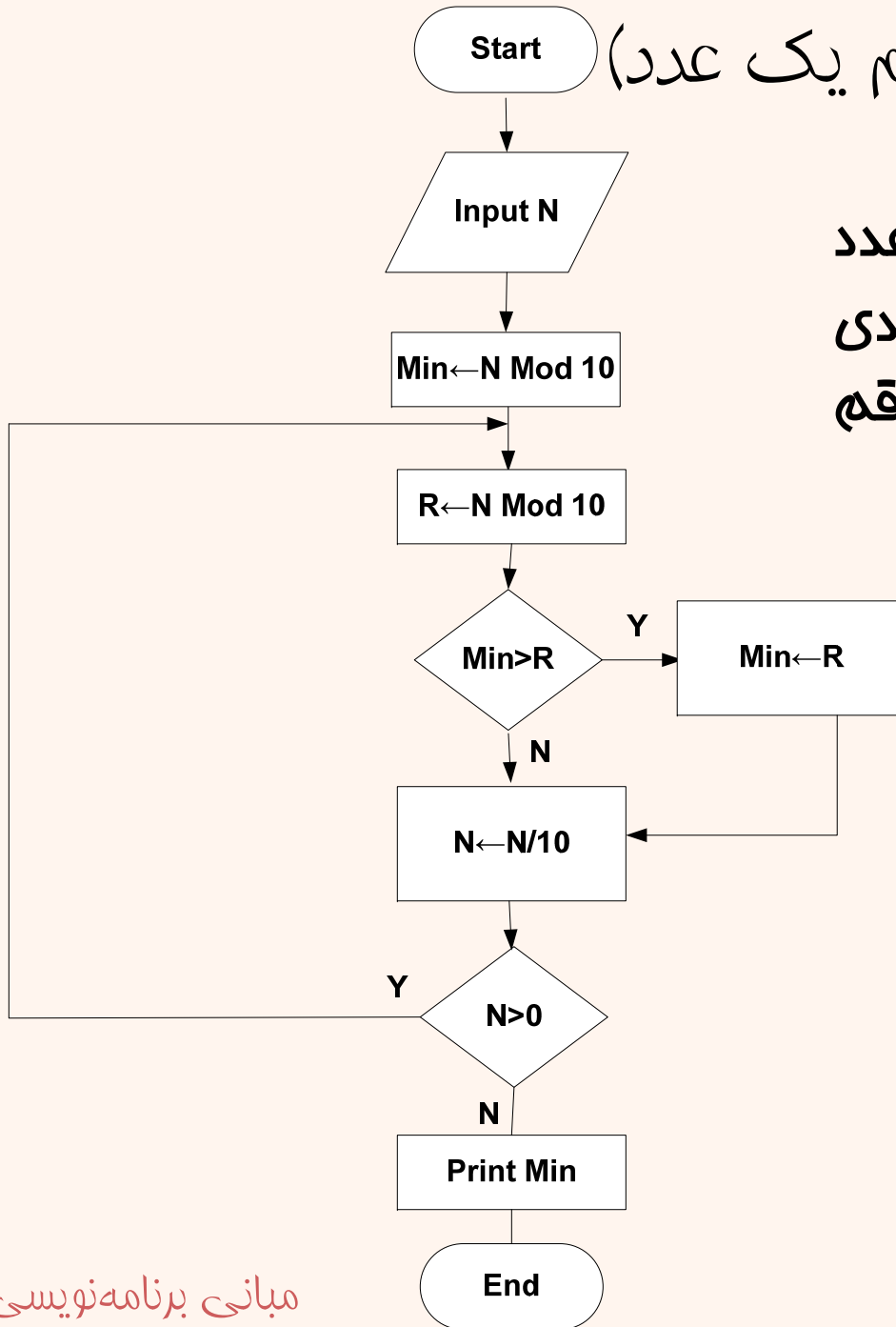
فهرست مطالب

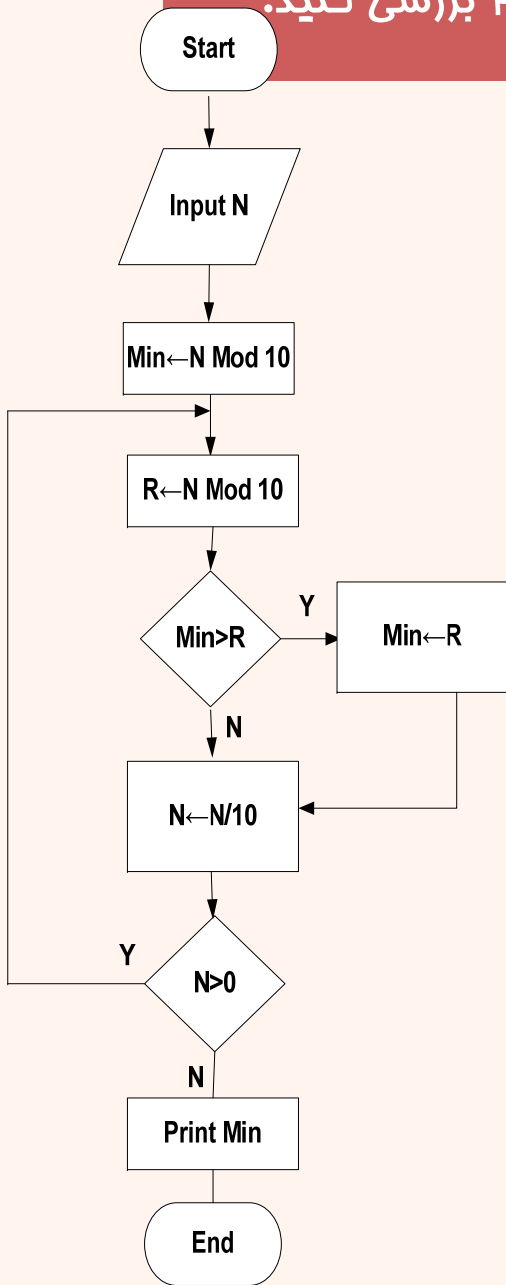
• روندنما (فلوچارت)



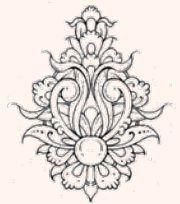
مثال (کوچک ترین رقم یک عدد)

- روندنمایی بکشید که عدد طبیعی N را از ورودی دریافت و کوچک ترین رقم آن را چاپ کند.



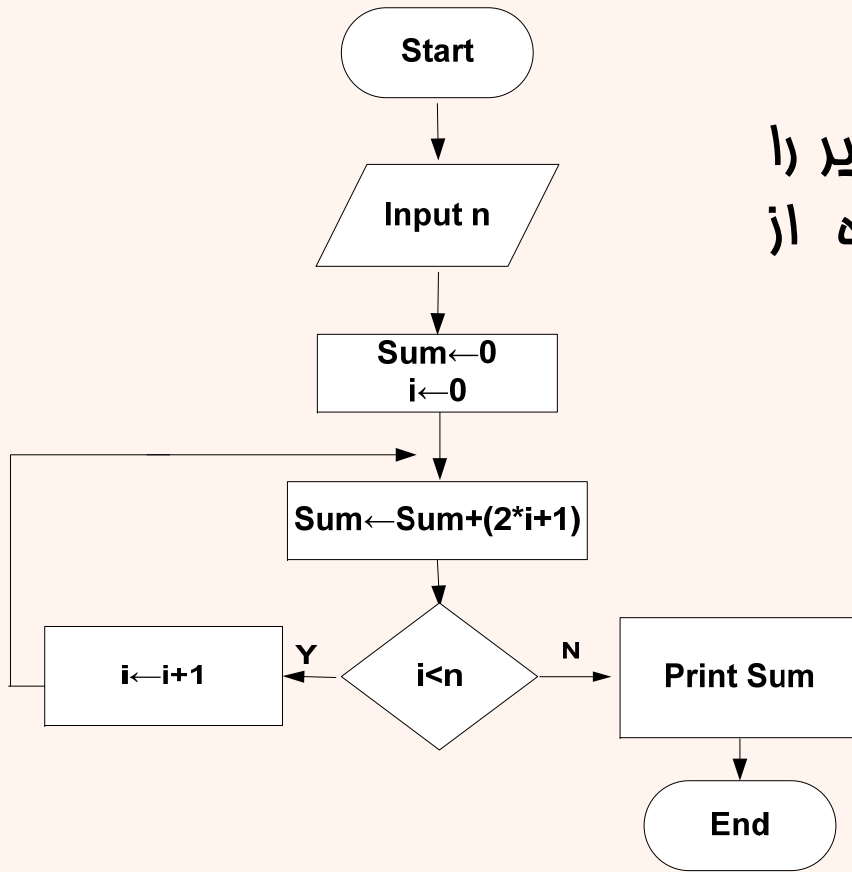


Min	R	N	
-	-	4172	Input N
2	-	4172	Min ← N Mod 10
2	2	4172	R ← N Mod 10
2	2	4172	Min > R
2	2	4172	Min ← R ❌
2	2	417	N ← N/10
2	2	417	N > 0
2	2	417	Print Min ❌
2	7	417	R ← N Mod 10
2	7	417	Min > R
2	7	417	Min ← R ❌
2	7	41	N ← N/10
2	7	41	N > 0
2	7	41	Print Min ❌
2	1	41	R ← N Mod 10
2	2	41	Min > R
1	1	41	Min ← R
1	1	4	N ← N/10
1	1	4	N > 0
1	1	4	Print Min ❌
1	4	4	R ← N Mod 10
1	4	4	Min > R
1	4	4	Min ← R ❌
1	4	0	N ← N/10
1	4	0	N > 0
1	4	0	Print Min

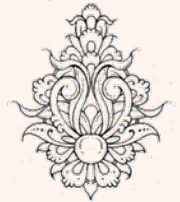


مثال (مجموع سری اعداد)

- روندنمایی بکشید که عبارت زیر را محاسبه کند. (بدون استفاده از فرمول تصاعد حسابی)



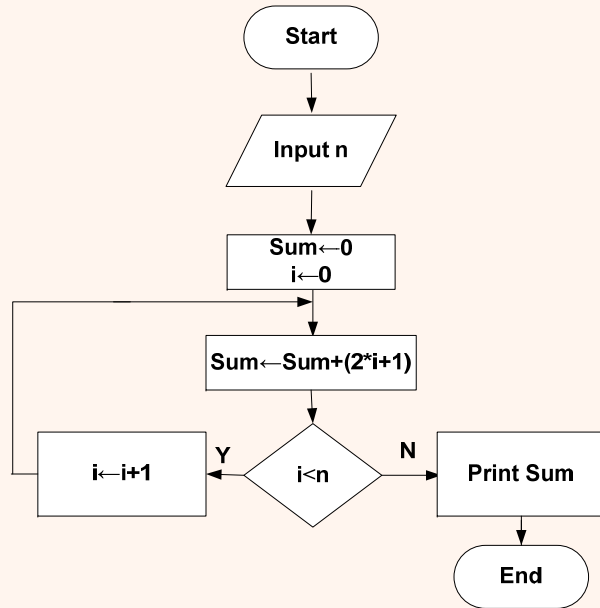
$$Sum = 1 + 3 + \dots + (2n + 1)$$



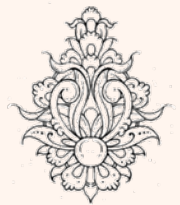
بررسی الگوریتم

• برای $n=2$ عبارت را حساب کنید.

$$Sum = 1 + 3 + \dots + (2n + 1)$$

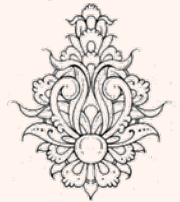
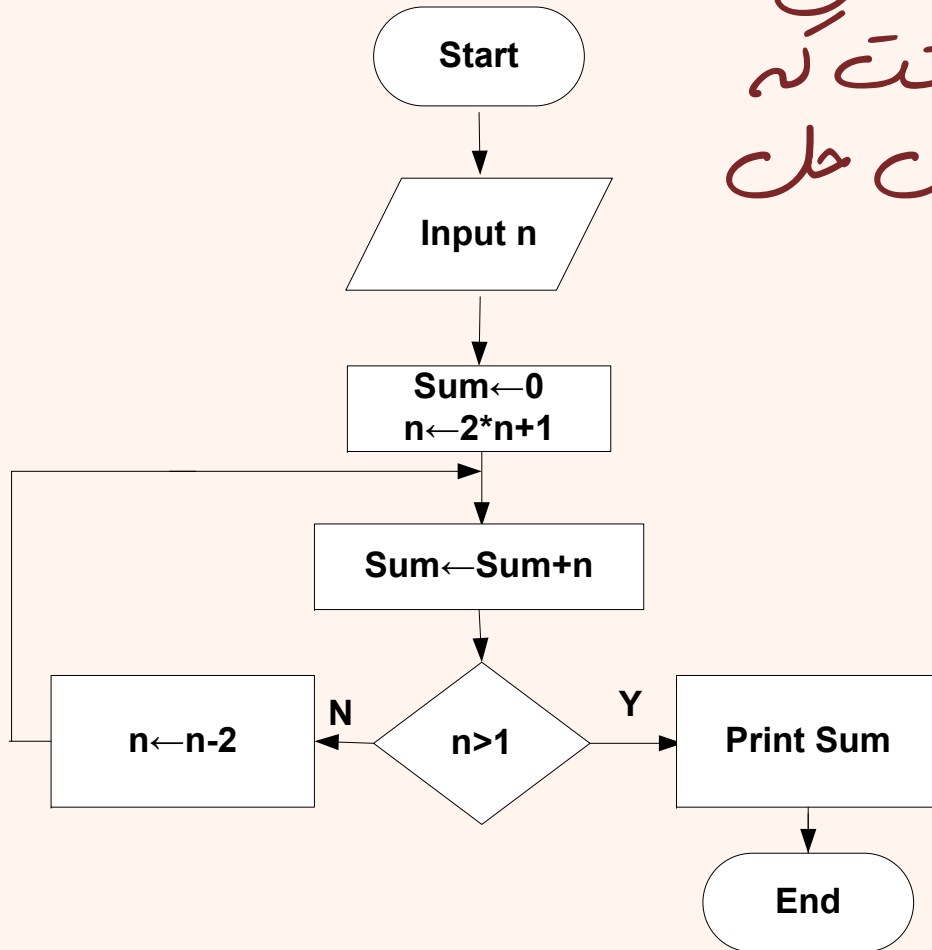


i	Sum	n	
-	-	2	Input n
0	0	2	Sum ← 0 i ← 0
0	1	2	Sum ← Sum + (2*i + 1)
0	1	2	i ≥ n N
1	1	2	i ← i + 1
1	1	2	Print Sum X
1	4	2	Sum ← Sum + (2*i + 1)
1	4	2	i ≥ n N
2	4	2	i ← i + 1
2	4	2	Print Sum X
2	9	2	Sum ← Sum + (2*i + 1)
2	9	2	i ≥ n Y
2	9	2	i ← i + 1 X
2	9	2	Print Sum



مثال (مجموع سری اعداد، ادامه...)

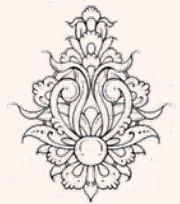
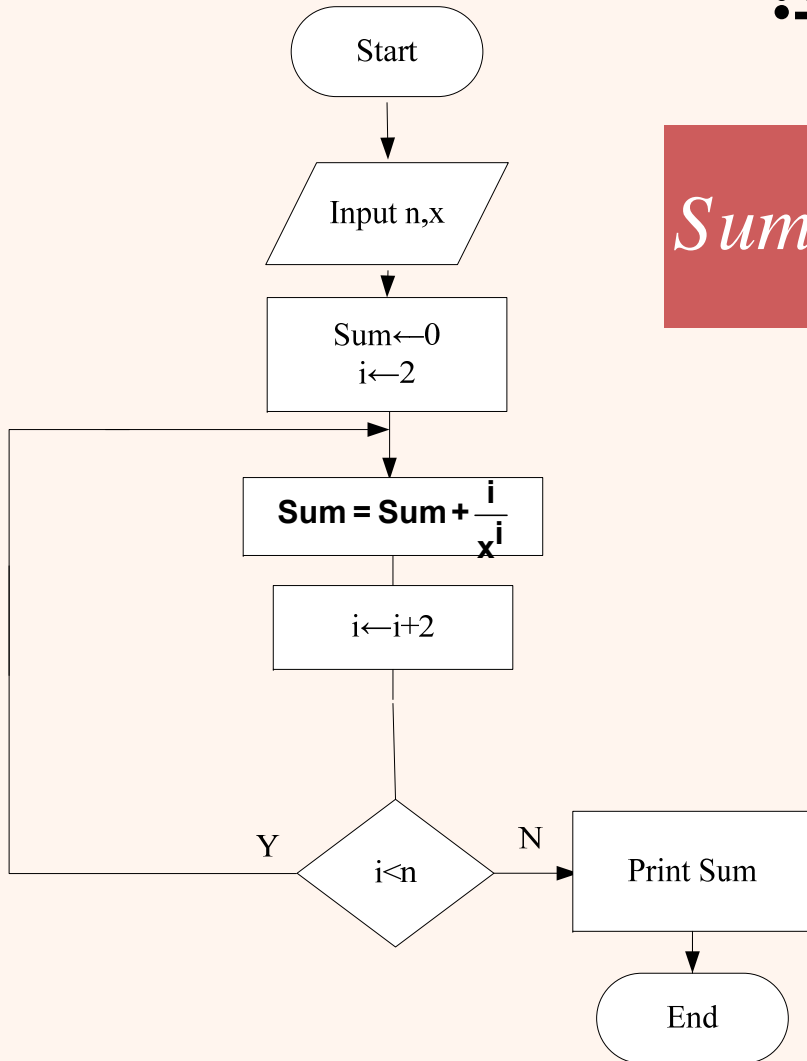
آیا می‌توانیم، روندنمای مثال
قبل را به لونه‌های نوشتن که
تنها با دو متغیر م‌آله قابل حل
باشد؟



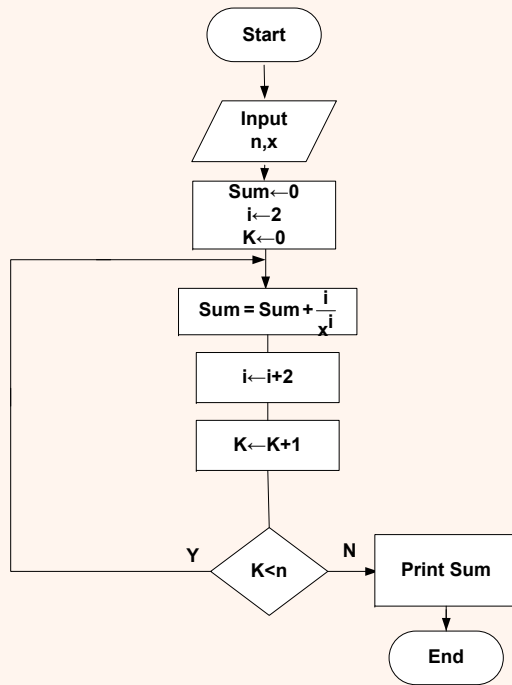
مثال (مجموع سری اعداد)

- الگوریتم و فلوچارتی بنویسید که n و x را بگیرد و عبارت زیر را محاسبه کند:

$$Sum = \frac{2}{x^2} + \frac{4}{x^4} + \frac{6}{x^6} + \dots + \frac{2n}{x^{2n}}$$

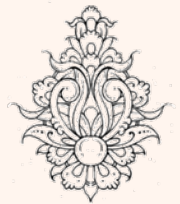


مثال (مجموع سری اعداد، ادامه...)



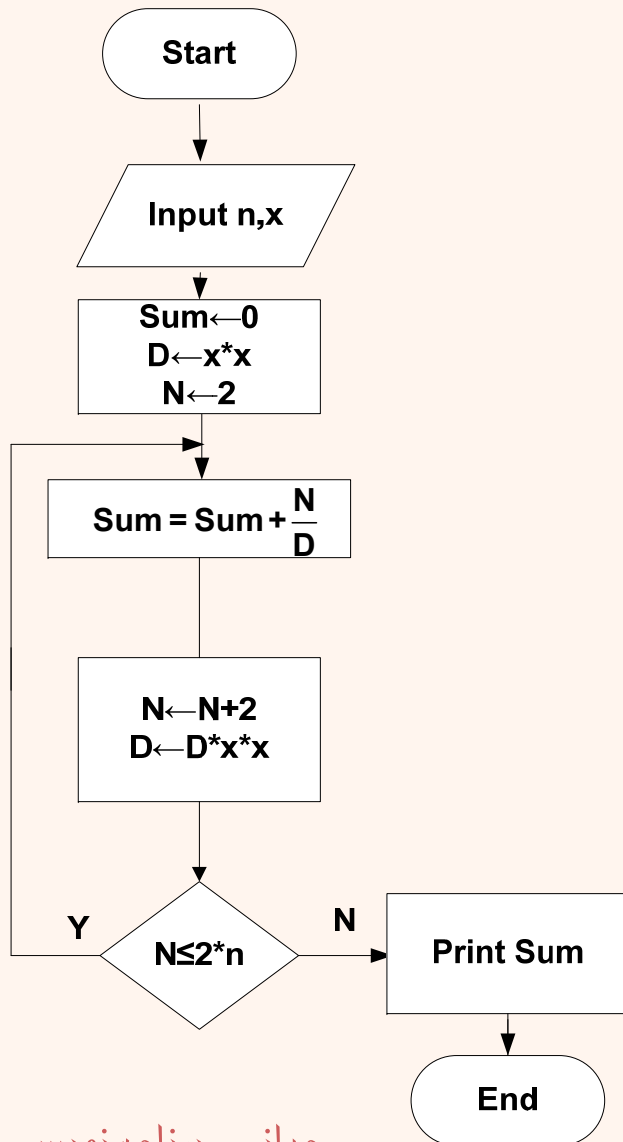
$$Sum = \frac{2}{x^2} + \frac{4}{x^4} + \frac{6}{x^6} + \dots + \frac{2n}{x^{2n}}$$

1. شروع
2. عدد x , n را دریافت کن
3. $Sum \leftarrow 0$, $i \leftarrow 2$ و $k \leftarrow 0$
4. $Sum \leftarrow Sum + i/x^i$
5. $i \leftarrow i+2$
6. $K \leftarrow K+1$
7. اگر $K < n$ برو به مرحله ی ۴
8. Sum را نمایش بده
9. پایان

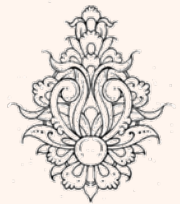


مثال (مجموع سری اعداد، ادامه...)

- در مثال قبل، از عملگر توان استفاده شده است، در صورت در دسترس نبودن این عملگر می‌توان فلوچارت را به صورت زیر رسم نمود:



- کارایی کدام شیوه بهتر است؟؟
- جای حلقه‌ی شرط را می‌توان جابجا کرد؟
- در این صورت چه تفاوتی خواهد داشت؟



انواع عملگرها

• عملگرهای دوتایی

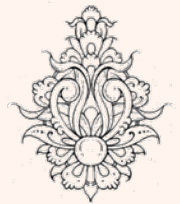
Binary Arithmetic Operators

Operator	Significance
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Remainder

• عملگرهای یکانی

Unary Arithmetic Operators

Operator	Significance
+ -	Unary sign operators
++	Increment operator
--	Decrement operator





• انتساب مرکب

```
i += 3;      → i = i + 3;  
i *= j + 2; → i = i * (j+2);
```




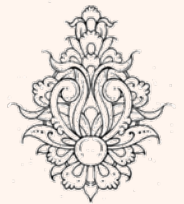
a -= c;  **a = a-c;**

a += 5;  **a = a+5;**

a *= 6;  **a = a*6;**

a /= b;  **a = a/b;**

a %= k;  **a = a%k;**



انواع عملگر (ادامه...)

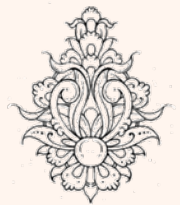
• عملگرهای نسبی

Relational Operators

Operator	Significance
<	less than
<=	less than or equal to
>	greater than
>=	geater than or equal to
==	equal
!=	unequal

• مثال

Comparison	Result
$5 \geq 6$	
$1.7 < 1.8$	
$4 + 2 == 5$	
$2 * 4 != 7$	

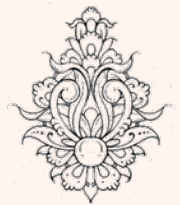


اولویت‌ها

Precedence	Operator
High ↑ ↓ Low	arithmetic operators
	< <= > >=
	== !=
	assignment operators

`bool flag = index < max - 1;`

• مثال



ابتدا معادلات صورت می‌گیرد، سپس مقایسه و در نهایت

انتساب

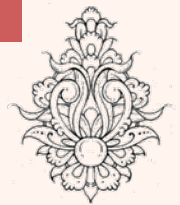


```
int result;  
result = length + 1 == limit;
```

ابتدا $length + 1$ محاسبه می‌شود و نتیجه با $limit$ مقایسه می‌شود و نهایتاً مقدار درست یا نادرست به $result$ منتسب می‌گردد.
چون $result$ عدد صحیح است صفر یا یک منتسب می‌گردد.

```
int result;  
(result = length + 1) == limit
```

ابتدا $length + 1$ محاسبه می‌شود و نتیجه در متغیر $result$ ریخته می‌شود. در نهایت مقدار به دست آمده با $limit$ مقایسه می‌شود و نهایتاً مقدار درست مقایسه می‌گردد.



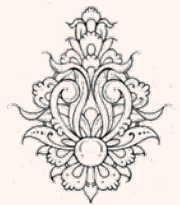
عملگرهای منطقی

Logical Operators

A	B	A && B	A B
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

A	!A
true	false
false	true

x	y	Logical Expression	Result
1	-1	<code>x <= y y >= 0</code>	
0	0	<code>x > -2 && y == 0</code>	
-1	0	<code>x && !y</code>	
0	1	<code>!(x+1) y - 1 > 0</code>	



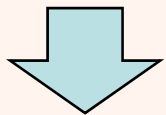
اولویت‌ها

Precedence	Operator
High ↑ ↓ Low	arithmetic operators
	< <= > >=
	== !=
	assignment operators

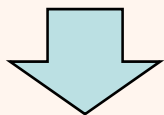
• اولویت در اپراتور های منطقی

اپراتور های یکسانی از بالاترین اولویت برخوردارند

مماسبات

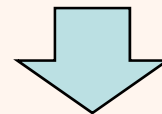


مقایسه

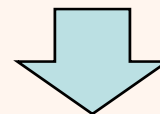


انتساب

نقیض !



&& و



یا ||

