

# برنامه‌سازی کامپیووتر

جلسه پنجم-بخش اول

ساختارهای تکرار در زبان C

# طرح کلی

---

- ساختار تکرار for

## مقدمه

---

- در حالت عادی دستورالعمل‌های برنامه‌ها بترتیب یکبار اجرا می‌شوند و برنامه خاتمه می‌یابد. اما در برخی موارد تعدادی از دستورات باید بیش از یکبار اجرا شوند. برای تکرار اجرای دستورالعمل‌ها از **ساختارهای تکرار استفاده** می‌شود.
- همچنین در برخی موارد برنامه با توجه به شرایط خاصی مسیرهای متفاوتی خواهد داشت. یعنی اجرای تعدادی از دستورالعمل‌های برنامه وابسته به شرایط مختلف متفاوت خواهد بود. برای بررسی این شرایط و انتخاب دستورالعمل‌های مناسب از **ساختارهای تصمیم استفاده** می‌شود.
- مثال: تعیین عدد بزرگتر، قدرمطلق، تبدیل مبنای دهدھی به دودویی.

# ساختار تکرار for

- در این ساختار برای تکرار اجرای دستورات حلقه‌ای ایجاد می‌شود و یک یا چند دستور در داخل حلقه قرار می‌گیرند.
- معمولاً در مواردی که تعداد تکرار حلقه مشخص باشد از `for` استفاده می‌شود.
- در این ساختار از متغیری برای کنترل تعداد تکرار حلقه استفاده می‌شود که آن را شمارنده یا اندیس حلقه تکرار می‌نامیم. اندیس حلقه دارای یک مقدار اولیه است و در هر بار تکرار حلقه (اجرای دستورات حلقه) مقداری به آن اضافه می‌شود. این مقدار را که پس از هر بار اجرای دستورات حلقه به اندیس حلقه افزوده می‌شود گام حرکت می‌گوئیم.
- گام حرکت می‌تواند عددی اعشاری و صحیح، منفی یا مثبت و کاراکتری باشد.
- همچنین هر حلقه دارای شرطی است که تعیین می‌کند حلقه تا چه زمانی باید ادامه داشته باشد که به آن شرط حلقه می‌گوئیم. اگر شرط حلقه دارای ارزش درست باشد حلقه تکرار می‌شود، اما اگر ارزش شرط حلقه نادرست باشد حلقه به پایان خواهد رسید.

```
(گام حلقه ; شرط حلقه ; مقدار اولیه اندیس حلقه)  
for {  
    ; دستور ۱  
    ; دستور ۲  
    ....  
    ; دستور n  
}
```

## نکته

---

- نکته: اگر حلقه فقط یک دستور داشته باشد، نیازی به } و { نمی‌باشد.

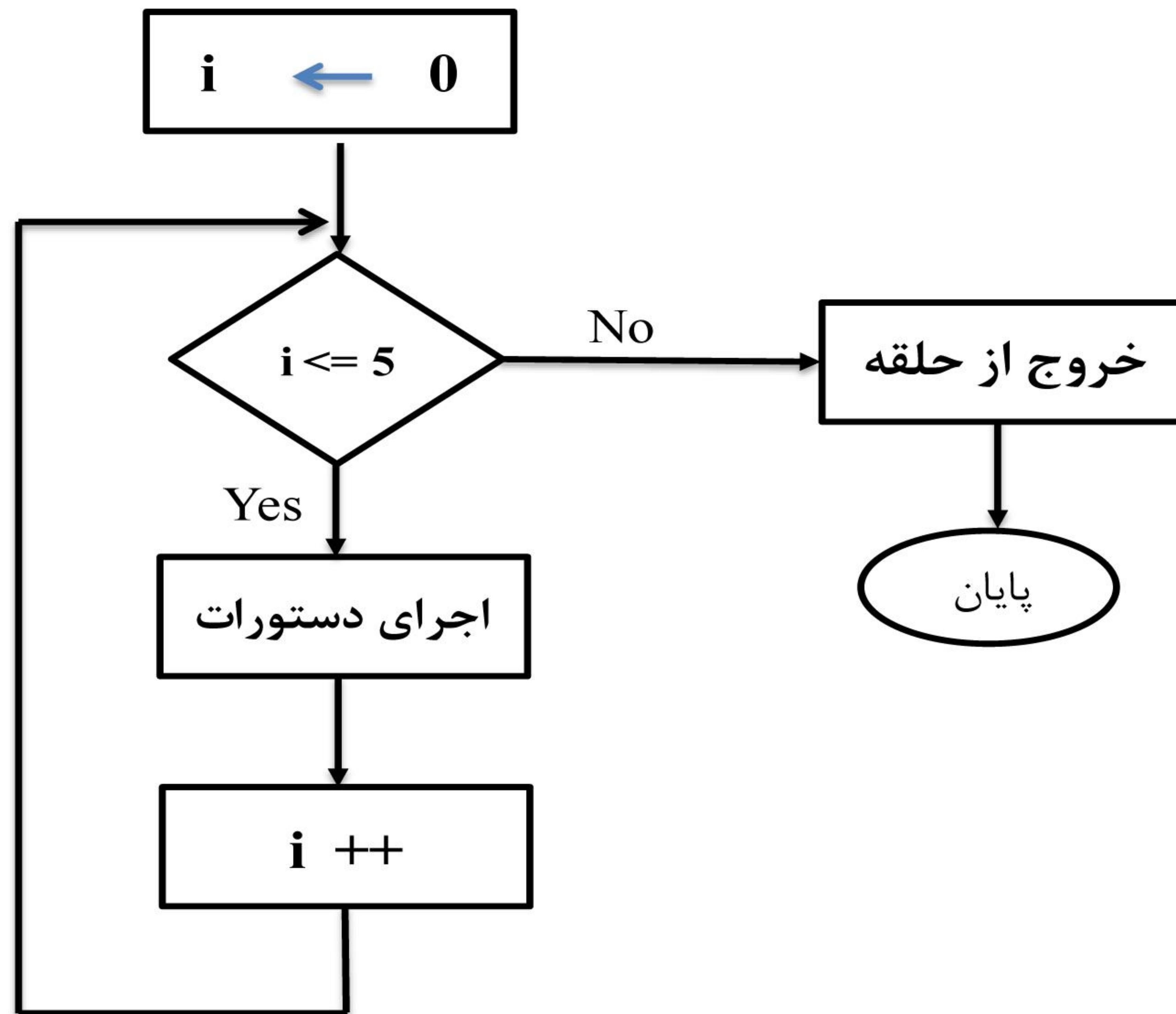
```
(گام حلقه ; شرط حلقه ; مقدار اولیه اندیس حلقه)  
for دستور 1
```

- نکته: (;;) for برای ایجاد حلقه بی‌نهایت مورد استفاده قرار می‌گیرد. حلقه بی‌نهایت فاقد شرط پایان است پس هیچ وقت متوقف نخواهد شد. در چنین موقعی برای توقف اجرای برنامه از صفحه کلید دکمه‌های Ctrl + Break مورد استفاده قرار می‌گیرند.

# مثال

---

```
for ( i = 0 ; i <= 5 ; i++ )  
printf( “%03d\n” , i);
```



# مثال

---

for (i = 1 ; i <= 10 ; i ++) printf( “ * ” );	* * * * *
for (i = 1 ; i <= 4 ; i ++) printf(“%d \n” , i );	1 2 3 4
for (i = 3.5 ; i <= 5 ; i +=0.5) printf (“%3.1f \n” , i );	3.5 4.0 4.5 5
for (i = ‘A’ ; i <= ‘F’ ; i +=2) printf(“%c \n” , i );	A C D F
int x=2, n=5; for (i = 1 ; i <= n ; i ++) x=2*x; printf(“%d”, x);	64

# مثال

---

برنامه‌ای بنویسید که اعداد مابین ۰.۵ تا ۳.۵ را با فاصله‌های ۰.۵ چاپ کند.

```
#include <stdio.h>
#include <conio.h>
```

```
int main()
{
    float i;
    clrscr();
    for (i = 0.5 ; i <= 3.5 ; i += 0.5)
        printf("%6.2f\n",i);
    getch();
    return 0;
}
```

متغیرها:

شمارنده حلقه

خروجی:

0.50

1.00

1.50

2.00

2.50

3.00

3.50

# مثال

برنامه‌ای بنویسید که کاراکترهای 'a' تا 'z' را به همراه کد اسکی آنها چاپ کند.

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
char ch;
```

```
for (ch = 'a' ; ch <= 'z' ; ch++)
```

```
    printf("ch = %c, code = %d\n" , ch , ch );
```

```
getch();
```

```
return 0;
```

```
}
```

متغیرها:

ch شمارنده کاراکترها

خروجی:

ch = a, code = 97

ch = b, code = 98

ch = c, code = 99

ch = d, code = 100

ch = e, code = 101

ch = f, code = 102

...

ch = z, code = 122

# مثال

برنامه‌ای بنویسید که پنج عدد صحیح را از ورودی خوانده و میانگین آن‌ها را محاسبه کرده و در خروجی نمایش دهد.

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int i, sum=0, num ;
    float ave;
    const int n = 5;
    clrscr();
    for (i = 1 ; i <= n ; i++)
    {
        printf("enter number %d : ", i);
        scanf("%d", &num);
        sum += num;
    }
    ave = (float) sum / n ;
    printf("\n the average is : %6.2f", ave);

    getch();
    return 0;
}
```

ثابت‌ها:

n ثابتی که نشانگر تعداد اعداد است.

متغیرها:

i شمارنده حلقه

sum مجموع اعداد

ave میانگین

num عددی که خوانده می‌شود

خروجی:

enter number 1: 12

enter number 2: 13

enter number 3: 14

enter number 4: 15

enter number 5: 16

the average is : 14.00

# حلقه‌های تکرار تودر تو

اگر حلقه تکراری داخل حلقه دیگری قرار گیرد، اصطلاحاً حلقه‌های توردت و گفته می‌شود. به ازای هر بار تکرار حلقه خارجی، حلقه داخلی به طور کامل اجرا می‌شود.

```
for ( i = 0 ; i <= 2 ; i ++) {  
    ....  
  
    for ( j = 1 ; j < 4 ; j++ ) {  
        ....  
    }  
    ....  
}
```

i	j
0	1
	2
	3
1	1
	2
	3
2	1
	2
	3

# مثال

---

```
for ( i = 1 ; i <= 8 ; i ++) {  
    for ( j = 1 ; j <= 5 ; j++ ) {  
        printf("*");  
    }  
    printf("\n");  
}
```

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

?

\*  
\*\*  
\*\*\*  
\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

# مثال

---

```
for ( i = 1 ; i <= 8 ; i ++) {  
    for ( j = 1 ; j <= i ; j++ ) {  
        printf("*");  
    }  
    printf("\n");  
}
```

```
*  
**  
***  
****  
*****  
*****  
*****  
*****
```

# مثال

برنامه‌ای بنویسید که جدول ضرب اعداد ۱ تا ۱۰ را چاپ کند.

```
#include <stdio.h>
#include <conio.h>
```

```
int main()
{
    int i ,j ;
    clrscr();
    for (i = 1 ; i <= 10 ; i++)
    {
        for (j = 1 ; j <= 10 ; j++)
            printf("%4d", i*j);
        printf("\n");
    }

    getch();
    return 0;
}
```

متغیرها:

۱، ۲ شمارنده‌های حلقه‌ها

خروجی:

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

Press any key to continue . . .

# حلقه‌های تکرار بی‌نهایت با استفاده از for

برنامه‌ای بنویسید که تعدادی عدد از ورودی خوانده و فاکتوریل آنها را محاسبه کند. تعداد اعداد معلوم نمی‌باشد و با زدن Ctrl+Break برنامه متوقف خواهد شد.

```
#include <stdio.h>
int main()
{
    long int fact;
    int i , num ;
    clrscr();
    for( ; ; )
    {
        printf("\nEnter a number : ");
        scanf("%d" , &num );
        fact = 1 ;
        for (i = 1 ; i <= num ; i++)
            fact *= i ;
        printf("Factorial is : %ld", fact );
    }
    getch();
    return 0;
}
```

متغیرها:

i شمارنده حلقه تکرار

num عدد خوانده شده

fact فاکتوریل عدد

خروجی:

Enter a number : 5

Factorial is : 120

Enter a number : 8

Factorial is : 40320

# عملگر کاما

---

- نکته: عملگر کاما انعطاف‌پذیری بیشتری به حلقه‌ی `for` می‌بخشد. با استفاده از این عملگر می‌توان در قسمت‌های مقدار اولیه و گام حرکت، دو یا چند عبارت را با هم ترکیب کرد.
- عبارات به ترتیب قرار گرفتن و از چپ به راست ارزیابی می‌شوند.
- مثال

```
for ( i = 0 , m += i ; i < 10 ; i++ , m++ )
```

# مثال

برنامه‌ای بنویسید که مجموع چند دوره اولیه از سری زیر را محاسبه کند.

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \dots$$

```
#include <stdio.h>
#include <conio.h>
#define num 6
int main()
{
    int count;
    float x , sum ;
    for( sum = 0 , x = 1.0 , count = 1 ; count <= num ; count ++ , x *= 2)
    {
        sum += 1 / x ;
        printf("sum = %7.4f, when count = %d\n", sum, count );
    }
    getch();
    return 0;
}
```

ثابت‌ها:

num برابر با تعداد جملاتی که باید جمع زده شوند.

متغیرها:

تعداد جمله در هر بار count

مجموع جملات در هر بار sum

مولد مخرج کسر x

خروجی:

sum = 1.0000, when count = 1

sum = 1.5000, when count = 2

sum = 1.7500, when count = 3

sum = 1.8750, when count = 4

sum = 1.9375, when count = 5

sum = 1.9688, when count = 6

# پایان بخش اول - جلسه پنجم