CS & IT ENGINEERING

Control flow statements

Iterative Statements -02

Lecture No. 03

By- Pankaj Sharma sir

1-for loop
2-While loop
3-Do while loop

1.

```
for (i=1; i<=3; i++)
{
    code
}
```

How many times code will execute

i=1,2,3 ⇒ 3 times

i=1 code
i=2 code
i=3 code

2.

```
for (i=1; i<=10; i++)
{
    code
}
```

i → 1,2,3,4,5,6,7,8,9,10 ⇒ 10 times

3.

```
for (i=1; i<=n; i++)
{
    code
}
```

n times

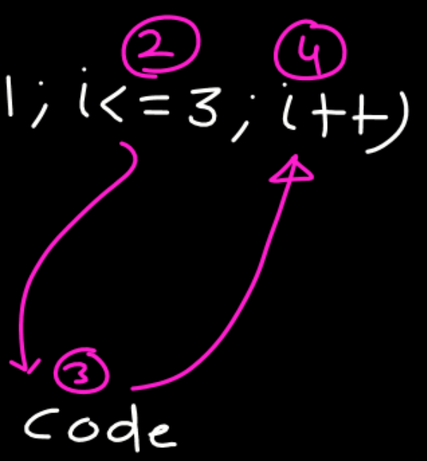
4.

```
for(j=1; j<=4; j++)
    {
      printf("Pankaj");
    }
```

Code is : To print Pankaj
4 times

5.

```
for(i=1; i<=3; i++)
    {
```




```
    }
```

i = 1 , 1<=3 True → code will execute
i = 2 , 2<=3 True → code will execute
i = 3 , 3<=3 True → code will execute

4.

```
for(j=1; j<=4; j++)
    {
      printf("Pankaj");
    }
```

Code is : To print Pankaj
4 times

5.

```
for(i=1; i<=3; i++)
    {
      for(j=1; j<=4; j++)
          {
            printf("Pankaj");
          }
    }
```

① ② ③

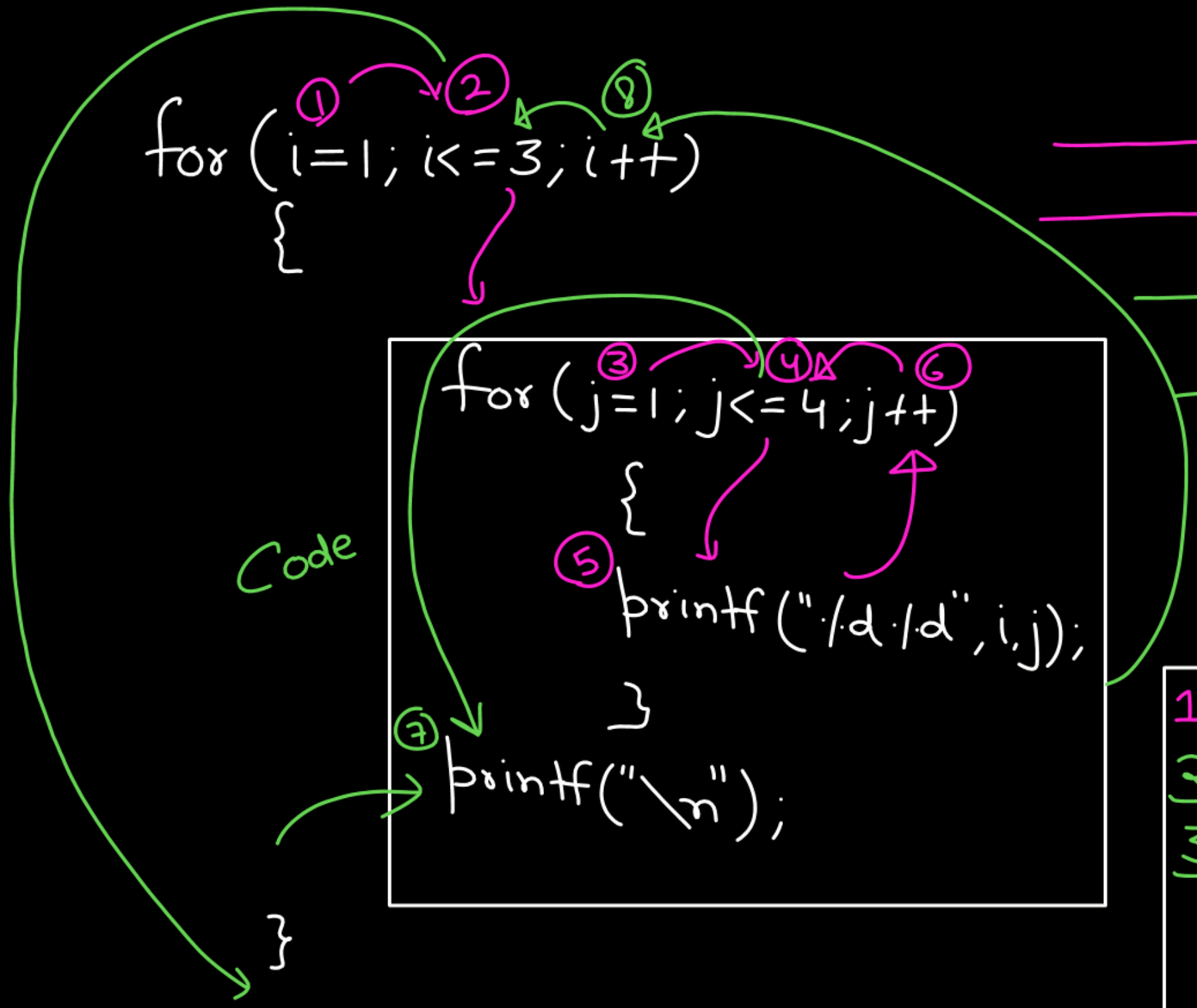
code

i=1 , 1<=3 True → code will execute 4 times Print Pankaj

i=2 , 2<=3 True → code will execute 4 times Print Pankaj

i=3 , 3<=3 True → code will execute 4 times

Q



i	condition	
1	1<=3 → True	j=1,2,3,4,5 ✗
2	2<=3 → True	j=1,2,3,4,5 ✗
3	3<=3 True	j=1,2,3,4,5 ✗
4	4<=3 False	

```
11121314
21222324
31 32333 4
```

Q

```
for (i=1; i<=3; i++)
{
    for(j=1; j<=4; j++)
    {
        printf("Pankaj");
    }
}
```

① ② ③ ④

i<=3 False

Code

i	j
1	1,2,3,4
2	1,2,3,4
3	1,2,3,4

i → Every value of i

⇒ inner loop

⇒ 4 times printf

i=1 → 4

i=2 → 4

i=3 → 4

4+4+4 (3 times)

= 3×4

Q

```
for (i=1; i<=3; i++)
    {
        for (j=1; j<=10; j++)
            {
             printf("Pankaj");
            }
    }
```

3 times

10 times

10+10+10 ⇒ 3×10

Q

```
for(i=1; i<=n; i++)
{
    for(j=1; j<=3; j++)
    {
        printf("Pankaj");
    }
}
```

n

3 times

$i = 1 \longrightarrow 3$

$i = 2 \longrightarrow 3$

$i = 3 \longrightarrow 3$

$\vdots$

$i = n \longrightarrow 3$

$3 + 3 + 3 + \cdots + 3$ (n times)

$= 3n$

Q

```
for (i = 1; i<=n; i++)        → n
    {
      for(j=1; j<=n; j++)     → n
          {
            printf("Pankaj");
          }
    }
```

i=1 ⟶ n
i=2 ⟶ n
i=3 ⟶ n
⋮
i=n ⟶ n

n + n + n + - - + n (n times)

= n × n

= n^2 times

①
```
for( j=1; j<=n; j++)
    printf("Pankaj");
```

for Compiler

```
for( j=1; j<=n; j++)
  {
    printf("Pankaj");
  }
```

②
```
for( i=1; i<=n; i++) {
    for( j=1; j<=n; j++)
      {
        printf("Pankaj");
      }
  }
```

Q

```
for (i=1; i<=n; i=i*2)
{
  printf("Pankaj");
}
```

$1+\lfloor \log_2 n \rfloor$

Q

```
for (i=1; i<=n; i++)
{
  for (j=1; j<=n; j=j*2)
  {
    printf("Pankaj");
  }
}
```

n

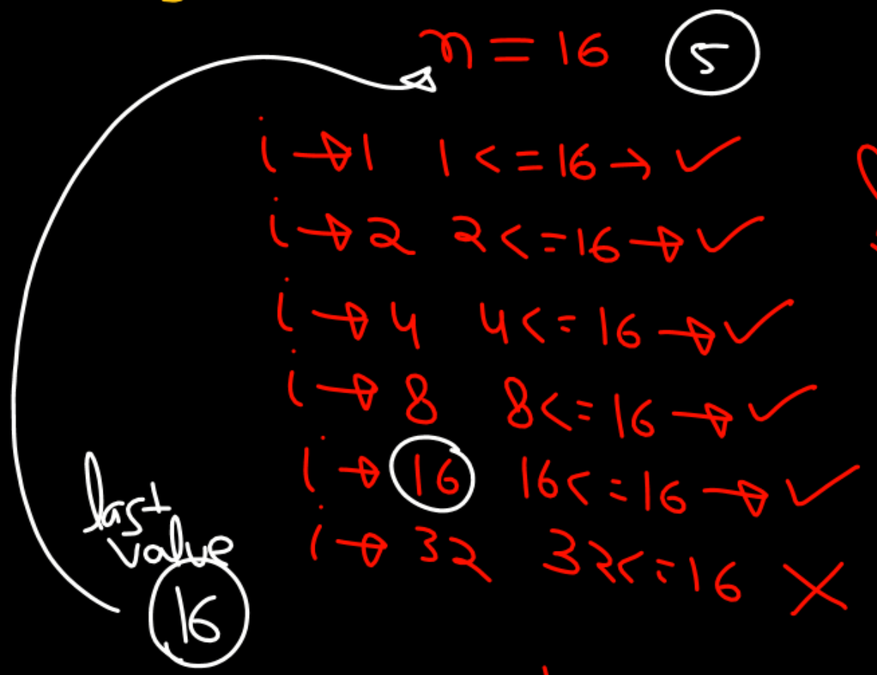
$1+\lfloor \log_2 n \rfloor$

$n(1+\lfloor \log_2 n \rfloor)$

```
for(i=1; i<=n; i=i*2)
    {

        printf("Pankaj");

}
```



n = 100

i → 1	1<=100	✓
i → 2	2<=100	✓
i → 4	4<=100	✓
i → 8	8<=100	✓
i → 16	16<=100	✓
i → 32	32<=100	✓
i → 64	64<=100	✓
i → 128	128<=100	✗

last value 64

```
for(i=1; i<=n; i= i*2)
  {
    printf("Pankaj")
  }
```

$(1+\lfloor \log_2 n \rfloor)$ times

$i = 1, 2^1, 2^2, 2^3, \ldots, 2^K$

Counting ?

$(K+1)$ times

$2^K <= n$

$\log(2^K) <= \log n$

$K \log 2 <= \log n$

$K <= \log n$

$K <= \frac{\log n}{\log 2}$

$K <= \log_2 n$

$\log a^b = b \log a$

$\frac{\log a}{\log b} = \log_b a$

Q

```
for (i=2; i<=100; i=i*2)
    {
     printf ("Pankaj");
    }
```

$i \to 2$ ✓
$i \to 2^2 \Rightarrow 4$ ✓
$i \to 8$ ✓
$i \to 16$ ✓
$i \to 32$ ✓
$i \to 64$ ✓
} 6 times

$i \to 128$ $128 <= 100$ ✗

Q

```
for (i=2; i<=n; i=i*2)
    {
     printf ("Pankaj");
    }
```

$i \to 2^1, 2^2, 2^3, \ldots 2^k$

k times

$2^k <= n$

$k = \lfloor \log_2 n \rfloor$

Q)

```
for (i=1; i<=3; i++)
    {
      for(j=1; j<=i ; j++)
          {
           printf("Pankaj");
          }
    }
```

i	j
1	1 → ①
2	1,2 → ②
3	1,2,3 → ③

1+2+3

= ⑥

1. $1+2+3+\cdots+k = \frac{k(k+1)}{2}$

2. $1^2+2^2+3^2+\cdots+k^2 = \frac{k(k+1)(2k+1)}{6}$

3. $1^3+2^3+3^3+\cdots+k^3 = \left[\frac{k(k+1)}{2}\right]^2$

A.P G.P

$a, a+d, a+2d, a+3d, \cdots$

$T_1 = a$

$T_2 = a+d$

$T_2 - T_1 = (a+d)-a = d$

$T_3 - T_2 = d$

$\vdots$

$$S_n = \frac{n}{2}\left[a_1 + l\right] \quad \text{(1)}$$

$$S_n = \frac{n}{2}\left[2a+(n-1)d\right] \quad \text{(2)}$$

$$S_n = \frac{a(r^n - 1)}{r - 1} \qquad \frac{a(1 - r^n)}{1 - r}$$

$$S_\infty = \frac{a}{1 - r}$$

dependent nested loop

```
for(i=1; i<=n; i++)
{
    for(j=i; j<=2*i; j++)
    {
        printf("Pankaj");
    }
}
```

i = 1	i = 2	i = 3	...
j = 1, 2	2, 3, 4	j = 3 to 6	
2	3	6-3+1 = 4	

i = n

j = n to 2n

2n - n + 1

n + 1

A.P

2 + 3 + 4 + ... + (n+1)

$\frac{1}{2}(1+3) \Rightarrow \frac{n}{2}(2+n+1)$

$= \frac{4}{2} = 2$

$= \frac{n}{2}(n+3)$

n = 1

2

9.

```
for(i=1; i<=n; i++)
  {
  for(j=i; j<=3*i; j++)
    {
    printf("Pankaj");
    }
  }
```

$i=1$	$i=2$	$i=3$	...	$i=n$
$j=1$ to 3	$j=2$ to 6	$j=3$ to 9		$j=n$ to $3n$
$3-1+1$	$6-2+1=5$	$9-3+1$		$\frac{3n-n+1}{2n+1}$
(3)	(5)	(7)		

$$3+5+7+\cdots(2n+1)$$

A.P

$a_1 = 3$

$N = n$

$l = 2n+1$

$n(n+2)$ times

$$S_N = \frac{N}{2}[a_1 + l]$$

$$= \frac{n}{2}[3+2n+1]$$

$$= \frac{n}{2}[2n+4]$$

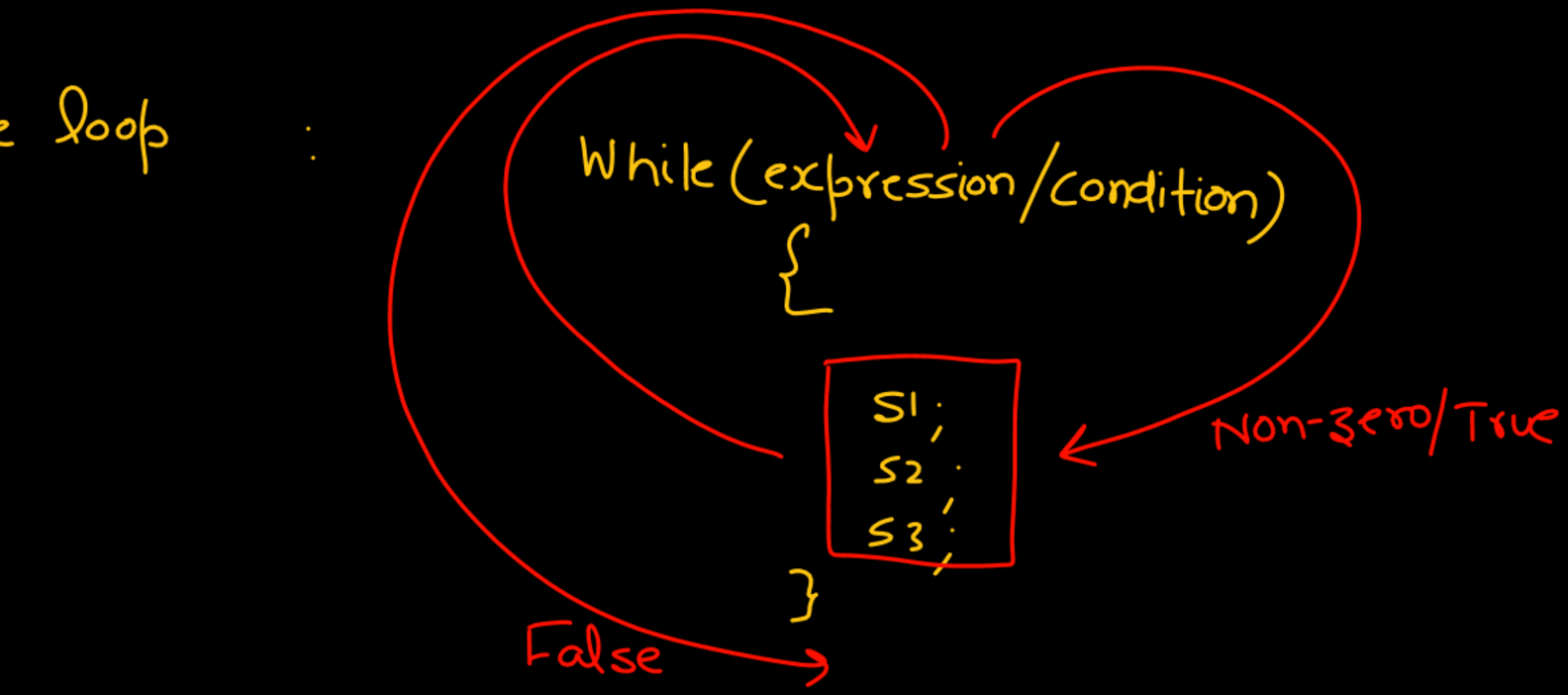
$$= \frac{n}{2} \cdot 2(n+2)$$

Q.
```
for (i = 1 ; i<=n; i = i*3)
    {
     for( j = i ; j<=n; j++)
          {
           printf ("Pankaj");
          }
    }
```

H.W

approx. ?

While loop :



```
for (i=1; i<=n; i++)
    {
     printf("Pankaj");
    }
```

```
i = 1;
While(i<=n){
     printf("Pankaj");
     i++;
     }
```

Q1:

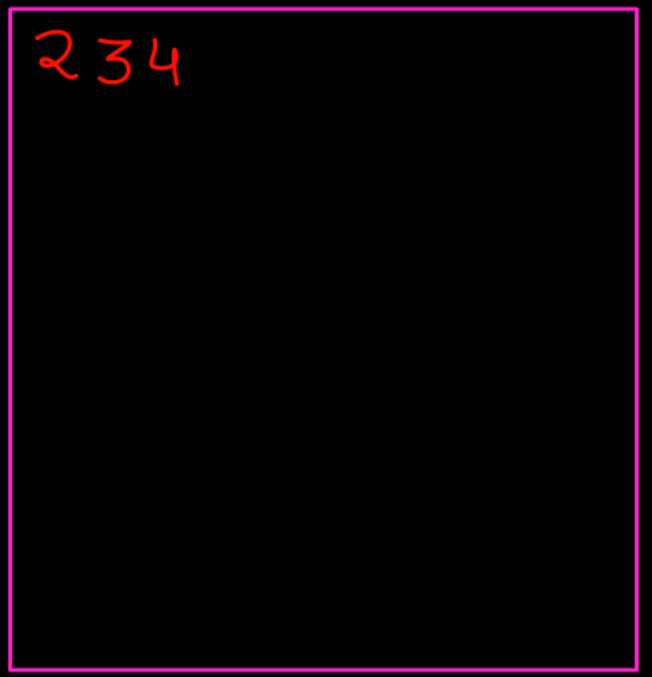
```
int i= 1;

while ( ++i< 5)
    {
      printf("%d", i);
    }
```

i | 1 ~~2~~ ~~3~~ ~~4~~ 5

2 < 5 → True
3 < 5 → True
4 < 5 → True
5 < 5 → False

O/P : 234



```
#include<stdio.h>
void main(){
    int i =1;
    while(++i<5);
    printf("%d",i);
}
```

First semi colon

```
int i=1;
while(++i<5)
{
;
}
printf("%d",i);
```

O/P :

~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

5

Mandatory

```
While( )
{
=
}
```

Compiler
Ud Ke
Laat Marega
— ???

1:

```
for(  ; 0 ; )
{
    printf("Pankaj");
}
```

condition

Pre-checking of condition

0 times

```
While (0)
{
    printf("Pankaj");
}
```

condition

0 times

do {

} while (expression);

do {

printf ("Pankaj");

} while (0);

False

1 time

atleast 1 time

for

While

① No. of iteration are known in advance

⇒ for

Table

```
for(i=1;i<=10;i++)
{
  ===
  ===
  ===
}
```

repeat

1 + 2 + 3 + 4 + 5

① Abtakkasum = 0 ;

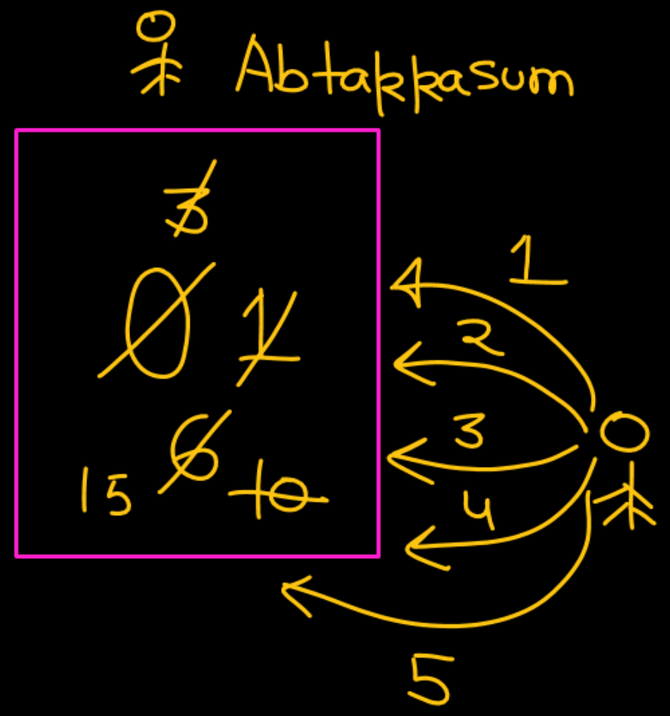
Abtakkasum = Abtakkasum + 1

Abtakkasum = Abtakkasum + 2

Abtakkasum = Abtakkasum + 3

Abtakkasum = Abtakkasum + 4

Abtakkasum = Abtakkasum + 5




```
Abtakkasum = 0;
for(i=1; i<=5; i++)
    Abtakkasum = Abtakkasum + i
```

```
Abtakkasum = 0;
for ( i=1; i<=5; i++)
    Abtakkasum = Abtakkasum + i ;
printf ("%d", Abtakkasum);
```

```
Abtakkasum = 0;
for ( i=1; i<=10; i++)
    Abtakkasum = Abtakkasum
                 + i;
printf ("%d", Abtakkasum);
```

```c
#include<stdio.h>
void main(){
    int n, i, sum;
    printf("Enter a number");
    scanf("%d", &n);
    sum = 0;
    for(i=1; i<=n; i++)
        sum = sum + i;
    printf("%d", sum);
    }
```

break ⇒

```
void main() { int i ;
for ( i=1; i<=10; i++)
{
    printf("Hello");
    break ;
}
printf("BYE");
}
```

1:

i	1

I<=10 True

HelloBYE

2:

```
for(i=1; i<=10; i++)
{
    break;
    printf("Pankaj");
}
```

<< endl

123

3:

```
for(i=1; i<=10; i++)
{
    (i) if(i%4==0)
        break;
    (ii) printf("%d", i);
}
```

1 2 3 4

1<=10 → True
2<=10 → True
3<=10 → True
4<=10 → True

1%4==0
1==0
2%4==0
2==0
3%4==0
4%4==0 True


THANK YOU
GW
SOLDIERS !