CS & IT ENGINEERING

Chapter 02

Programming in C

Control Flow Statements

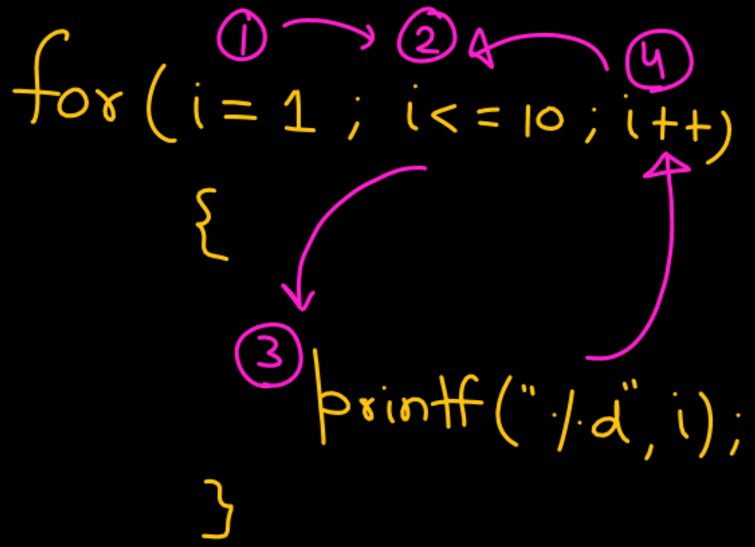
Lec- 05

By- Pankaj Sharma sir

More on Loops-1

Continue

loop ⇒ iteration



i	
1	1<=10 → 1
2	

O/P: 12345678910

1245 7810 →

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

    if(i%3==0)
      continue;
    printf("%d", i);
  }
```

i=7 → ✓
i=8 → ✓
i=9 → pf X
i=10 → pf ✓

i=4 → 4%3==0 → false
pf → 4
i=5 → 5%3==0 false
pf → 5
i=6 → 6%3==0 true

Continue: Skip the remaining code of current iteration and continue with next iteration.

i		
1	1<=10 →	1%3==0 false → continue X pf → 1
2	2<=10 →	2%3==0 → continue X pf → 2
3	3<=10 →	3%3==0 → true continue.

break

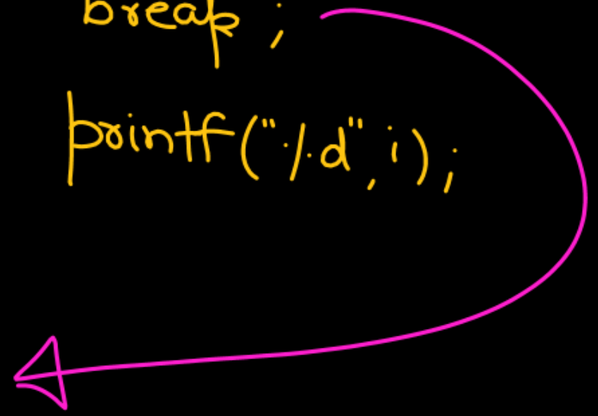
```
for(i=1; i<=10; i++)
  {
    printf("%d",i);
  }
```

i	
1	1%3==0 → false break X pf → 1
2	2%3==0 → false break X pf → 2
3	3%3==0 → true break loop terminate

```
for( i=1; i<=10; i++)
  {
    if( i%3==0)
        break;
    printf("%d", i);
  }
```

O/P 1 2

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



O/P : No output

11 31 32 41

```
for(i=1; i<=5; i++)
{
    for(j=1; j<=5; j++)
    {
        if((i+j)%3==0)
            break;
        printf("%d %d", i, j);
    }
}
```

i = 4 j=1 → 41
j=2 → break ✓

i = 5 j=1 → break

i	j	
1	1	(1+1)%3==0 → false break; ✗ 11 printed
	2	(1+2)%3==0 → true → break; → inner loop से बाहर
	j = 3,4,5 ✗	
2	1	(2+1)%3==0 → true → break → inner loop से बाहर
3	1	(3+1)%3==0 false break ✗ 31
	2	(3+2)%3==0 false break ✗ 32
	3	(3+3)%3==0 true break ✓

loop

1+2+3+4+5

Abtakkasum = 0

repeat

Abtakkasum = Abtakkasum + 1

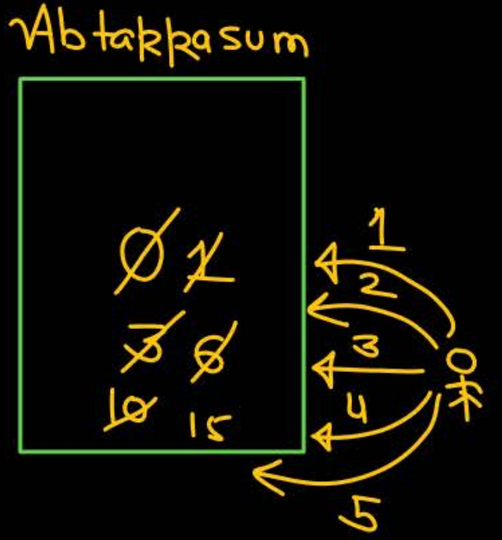
Abtakkasum = Abtakkasum + 2

Abtakkasum = Abtakkasum + 3

Abtakkasum = Abtakkasum + 4

Abtakkasum = Abtakkasum + 5

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



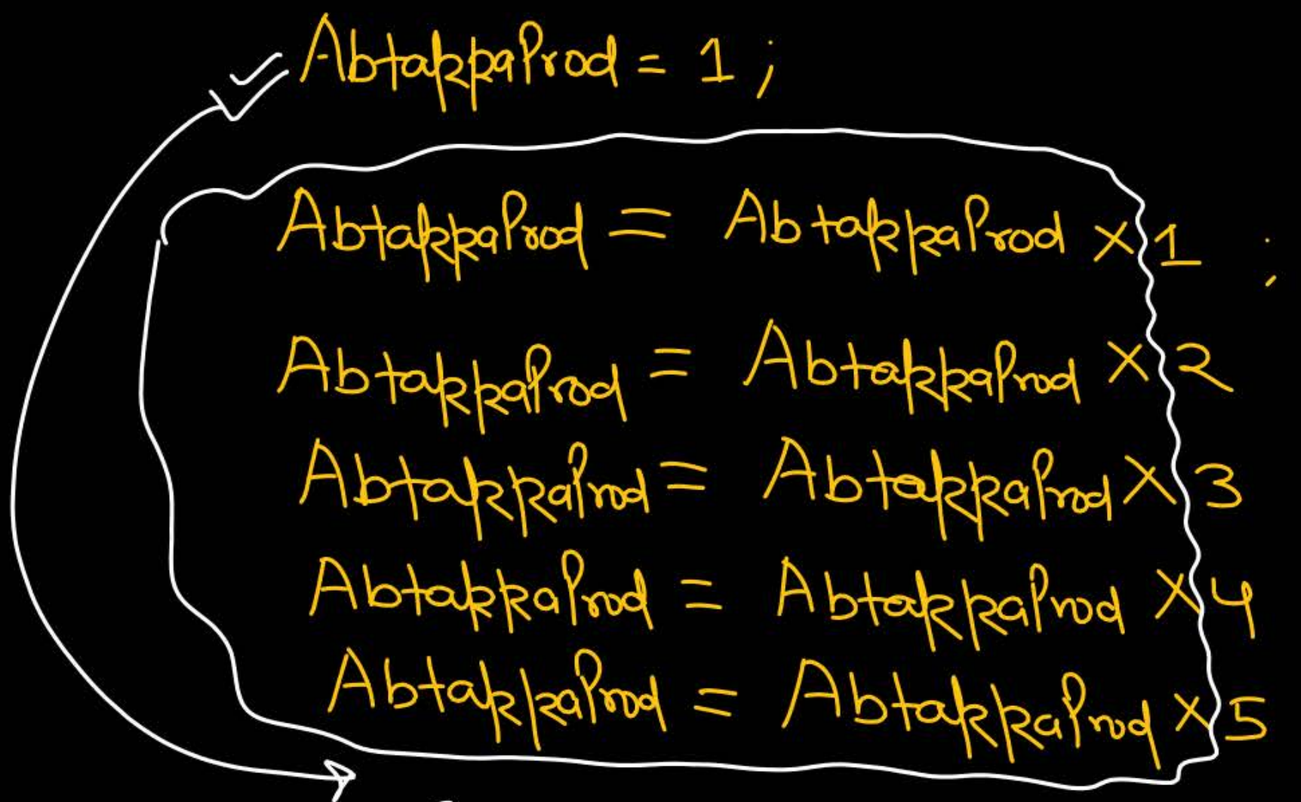
```
Abtakkasum=0
for(i=1;i<=5;i++)
    Abtakkasum = Abtakkasum + i
printf("%d", Abtakkasum);
```

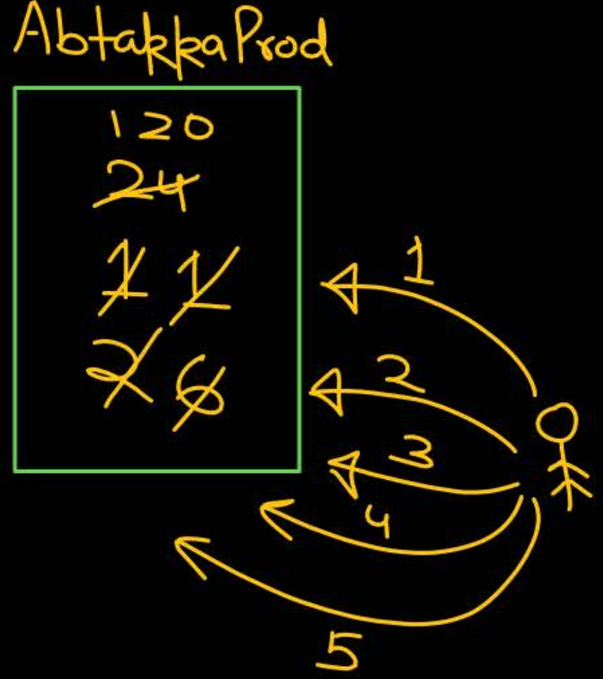
```
sum=0;
for(i=1;i<=10;i++)
        sum = sum+i;
printf("%d", sum);
```

i/p → n using scanf.

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

1×2×3×4×5






```
for(i=1; i<=5 ; i++)
    AbtakkaProd = AbtakkaProd × i ;
```

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

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

factorial

1.2.3...n

```
Prod = 1;
for(i=1; i<=n; i++)
        Prod = Prod * i;
printf("%d", Prod);
```

Perfect Number

6 = 1, 2, 3, ~~6~~

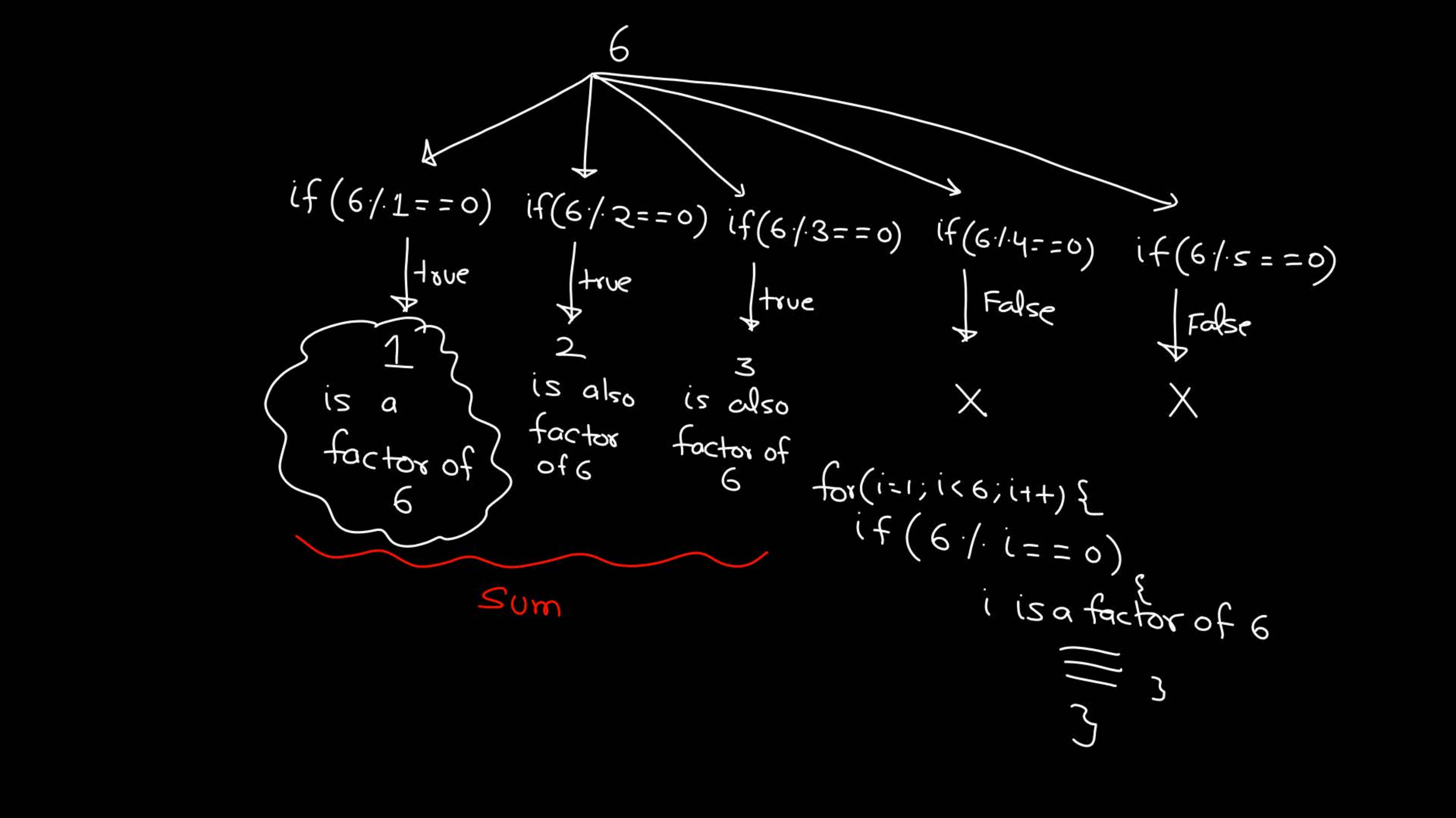
1+2+3 (sum of all factors other than the no. itself is equal to the no.)

$$6 = 1+2+3$$

28

1, 2, 4, 7, 14

1+2+4+7+14 = 28


6
if (6 % 1 == 0)
if(6 % 2 == 0)
if(6 % 3 == 0)
if(6 % 4 == 0)
if(6 % 5 == 0)
true
true
true
False
False
1
is a factor of 6
2
is also factor of 6
3
is also factor of 6
X
X
Sum
for(i=1; i<6; i++) {
if (6 % i == 0)
i is a factor of 6
{
}
3

Sum → factors का sum

n = 6

1, 2, 3

Sum = 6

```
sum = 0;
for (i=1; i<n ; i++)
    {
        if (n % i == 0)
            {
        // i is a factor of n
        Sum = Sum + i ;
            }
    }
if (sum == n)
    → perfect
else
    → Not perfect
```

i/p : 147
o/p : 3

i/p 1276
o/p 4

Count the no. of digits in a given no. n

147 ⟶

(i) 147/10 = 14
int int

(ii) 14/10 = 1

(iii) 1/10 = 0


if (n>0)
{
code
}
else
{
}

$a \geq 1$
$b \geq 1$

a^b

3^5

$3 \times 3 \times 3 \times 3 \times 3$

3^5

```
Prod = 1
for(i=1; i<=5; i++)
    Prod = Prod x 3;
```

```
Prod = 1
for(i=1; i<=5; i++)
    Prod = Prod * i;
```

$1 \times 2 \times 3 \times 4 \times 5$

$1 \times 2 \times 3 \times 4 \times 5$

$a \times a \times a \times \cdots$ a (b times)

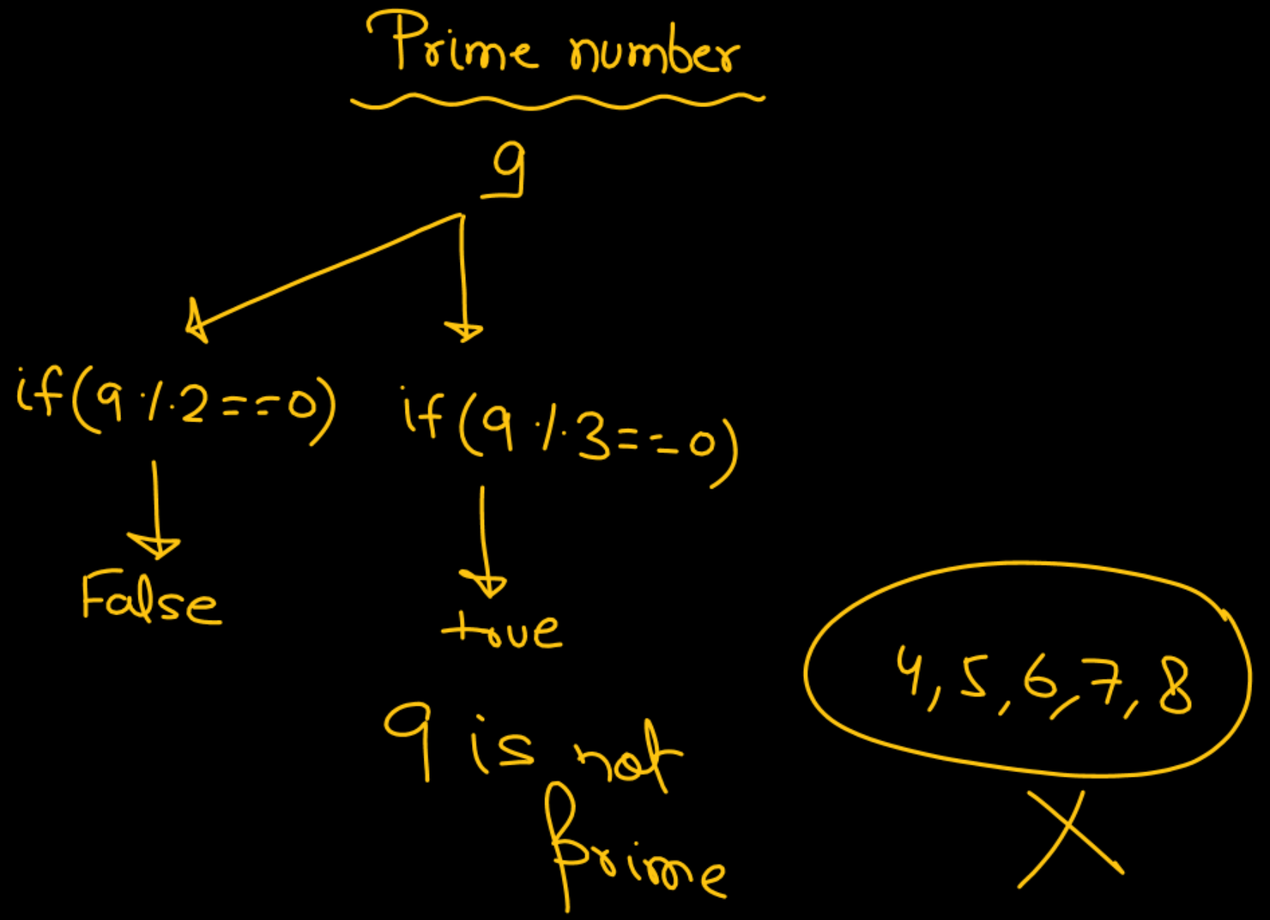
a^b

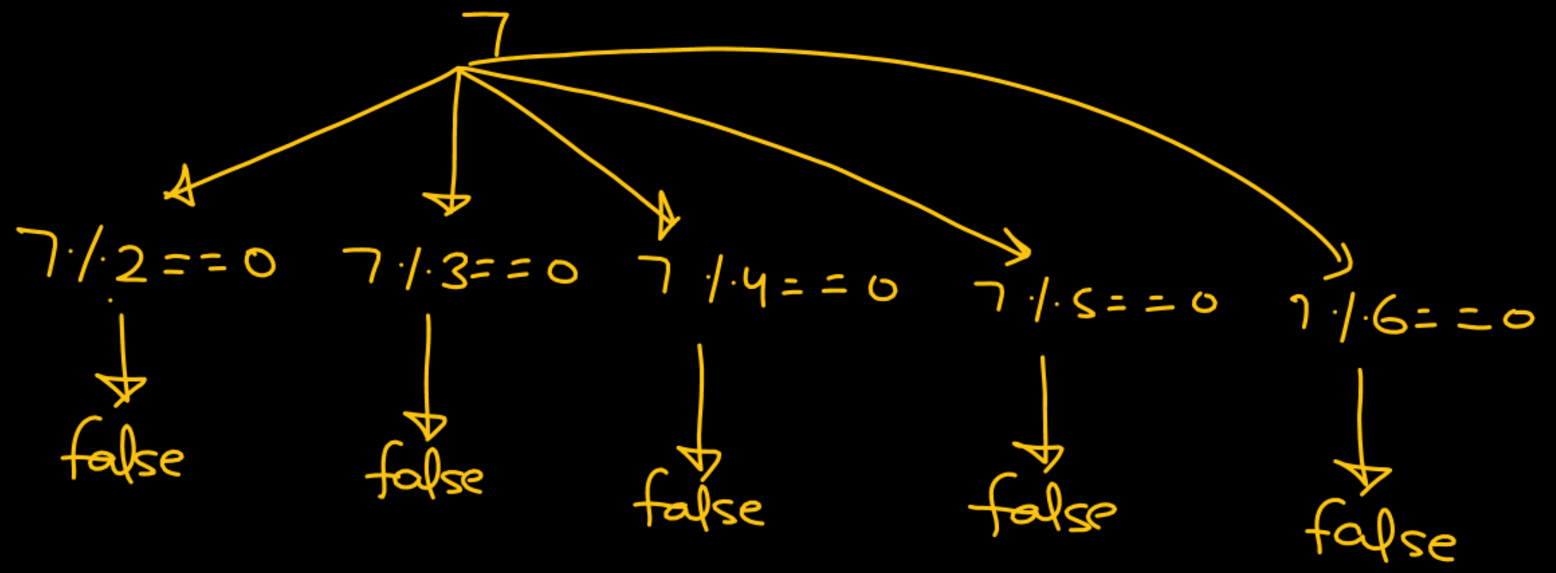
```
Prod = 1;
for( i = 1; i <= b; i++)
    Prod = Prod x a;
```

$\approx a^b$

Pow(. , .)

Prime number





7 → 2, 3, 4, 5, 6

```
for(i=2;i<7;i++){
    if(7%i==0)
    {
        some code
    }
}
```

n = 9

i	
2	9 % 2 == 0 → falsex
3	9 % 3 == 0 → true

```
for(i=2; i<n; i++)
    {
        if(n % i == 0)
            {
            printf("%d is not prime", n);
            break;
            }
    }
```

n = 7

i	
2	7 % 2 == 0 → x
3	7 % 3 == 0 → x
4	7 % 4 == 0 → x
5	7 % 5 == 0 → x
6	7 % 6 == 0 → x
(7)	

```
for(i=2; i<n; i++)        // 2<2
  {
     if(n%i==0)
       {
        printf("%d is not prime",n);
        break;
       }
  }
if(i==n)
    printf("%d is prime",n);
```

n=2

n==1

Special

Print the sum of digits of a number 2 min

$n > 0$

- Logic

I/P : 127

O/P : 10

I/P : 3965

O/P : 23

127

last = 127 % 10

last = 7

12347 % 10 = 7

1234 % 10 = 4

123 % 10 = 3

12 % 10 = 2

127 → 7

12 → 2

भाग और शेष

/ %

1234

1234 % 10 = 4

1234 / 10 = 123

1234 % 100 = 34

1234 / 100 = 12

$n = 127$

```
last = n % 10
n = n/10
```

(i) last = 127 % 10 = 7
n = 127/10 = 12

(ii) last = 12 % 10 = 2
n = 12/10 = 1

(iii) last = 1 % 10 = 1
n = 1/10 = 0

```
sum = 0;
while (n != 0)
  {
   last = n % 10;
    n = n/10;
   sum = sum + last;
  }
```

Armstrong number

- Easy version
- original version

$n = 153$

$1^3 + 5^3 + 3^3 = 1 + 125 + 27$

$= 153$

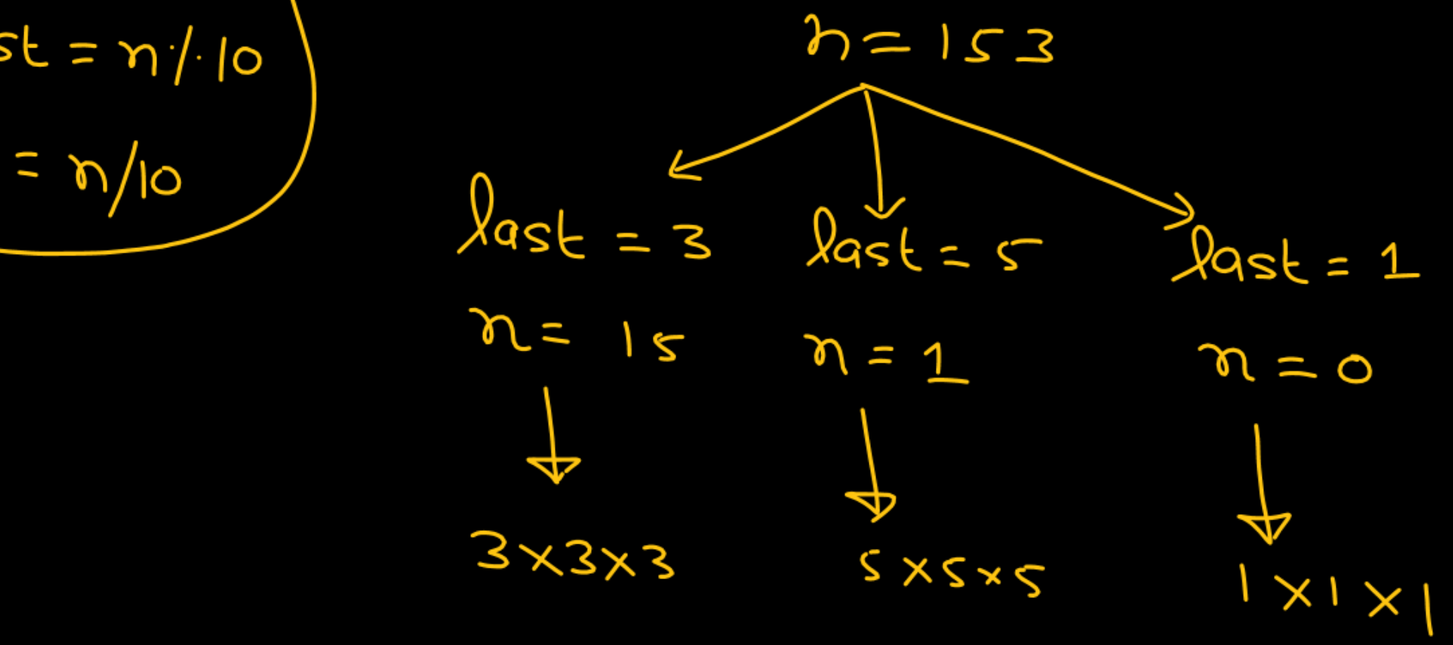
Sum of cubes of digits of a number

$=$ number

$n = 153$

last = n % 10

n = n/10



```
sum = 0;
while (n != 0)
  {
     last = n % 10;
      n = n/10;
sum = sum + last x last x last;
  }
```

this loop will make n as 0

Sum == n ✗ don't

```
a = n;
sum = 0;
while (n != 0)
   {
      last = n % 10;
       n = n/10 ;

sum = sum + last x last x last ;
   }
if ( sum == a)
        ⟶ Yes
else
        ⟶ No
```

$n = d_1 d_2 d_3 d_4$

$$d_1^{\text{no. of digits}} + d_2^{\text{no. of digits}} + d_3^{\text{no. of digits}} + d_4^{\text{no. of digits}}$$

(i) No. of digits counting ✓ $= n$

(ii) a^b ✓

(iii) Sum ✓

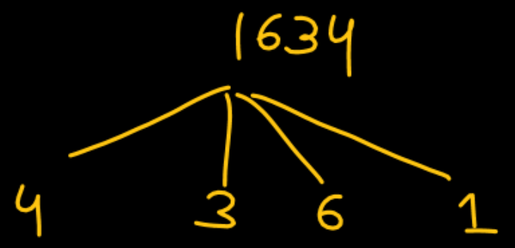
```
a=n;
count = 0;
while(n!=0){
        n= n/10;
        count++;
        }
```

after this code

n = 1634

Count = 4

n = 0




```
sum=0;
    while(a!=0)
    {
digit = a%10;
  a = a/10;
digit^count [
  Prod=1;
  for(i=1; i<=count; i++)
          Prod = Prod x digit;
sum = sum + Prod;
}
```

```
Prod = 1;
for(i=1; i<=b; i++)
    Prod = Prod x a;
```

```
b = a = n;
count = 0;
while(n!=0){
        n = n/10;
        count++;
        }
```

after this code

n = 1634

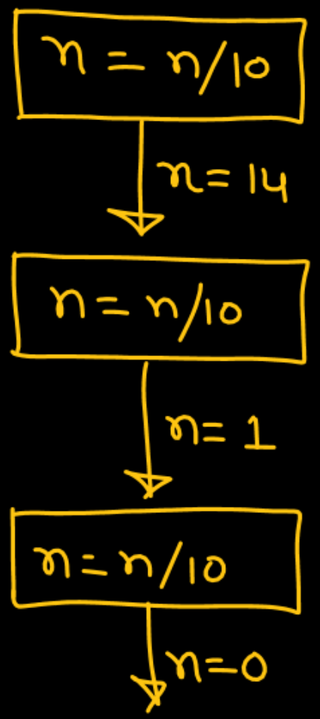
Count = 4
n = 0

```
sum = 0;
    while(a!=0)
  {
digit = a%10;
  a = a/10;
 Prod = 1;
 for(i=1; i<=count; i++)
        Prod = Prod x digit;
  sum = sum + Prod;
 }
if(sum<=b)
    printf("%d is Bahubali", b);
else
    printf("%d is not Bahubali", b);
```

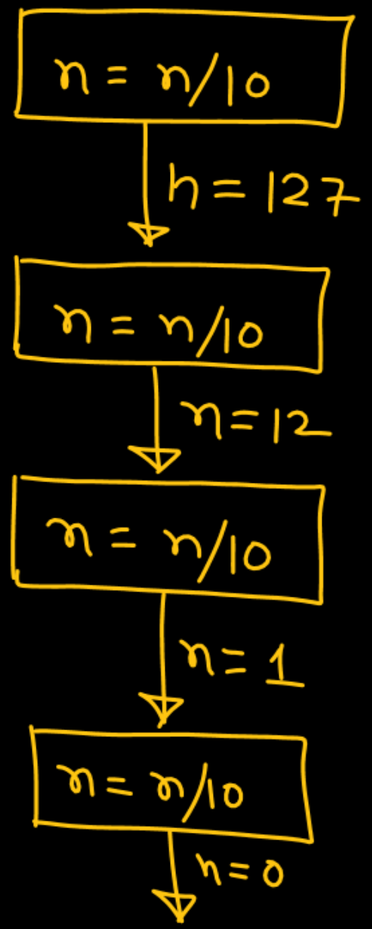
```
count = 0;
While(n!=0)
  {
    n = n/10 ;
    count ++ ;
  }
```

→ count →

n = 147



n = 1278




THANK YOU GW
SOLDIERS !