Loop for printing odd numbers less than 10

```
int i=1;
while (i <= 10) {
    SOP(i);
    i= i+2;
}
```

1 3 5 7 9 11 ↓ Breaks

Loop for balling all balls of an over

```
int count=1;
while (count <= 6) {
    // bowl a ball
    count ++;
}
```

Last digit of an integer N

Modulo %

1234 ⇒ 4

569 ⇒ 9

10 ⇒ 0

$(123 \times 10 + 4)/10$

$\Rightarrow 123 + \frac{4}{10} \Rightarrow$ 4 Remainder

$\frac{569}{10} \Rightarrow \frac{56 \times 10}{10} + \frac{9}{10} \Rightarrow 56 + \frac{9}{10}$

⇓

9 Remainder

$\frac{10}{10} \Rightarrow \frac{10+0}{10} \Rightarrow \frac{10}{10} + \frac{0}{10} \Rightarrow 1 + \frac{0}{10}$

0 Remainder

1234 % 10 = Remainder after division by 10

1234 % 10 = 4

569 % 10 = Remainder after / by 10
= 9

N % 10 ⇒ Gives last digit of N

Given a number print all the digits

N = 6341

SOP(N % 10) ⇒ 1
SOP (N % 100) ⇒ 41
SOP (N % 1000) ⇒ 341
SOP(N % 10000) ⇒ 6341

Output
1
4
3
6

n/10 6341 % 10 ⇒ 1
n/10 634 % 10 ⇒ 4
n/10 63 % 10 ⇒ 3
n/10 6 % 10 ⇒ 6

↘ 0 ⇒ Break the loop

```
int n = scn.nextInt();
while (n != 0) {
    SOP(n%10);
    n = n/10
}
```

n	condition	output	new n
6341	True	1	634
634	True	4	63
63	True	3	6
6	True	6	0
0	False ⇒	Break	

Input ⇒ 0
Output ⇒ 0

```
int n = scn.nextInt();
while (n != 0) {
    SOP(n%10);
    n = n/10
}
```

No output

```
int n = scn.nextInt();
while (n >= 0) {
    SOP(n%10);
    n = n/10
}
```

0

Output : 0 0 0 0 0 0 .... infinite times

Infinite Loop

```
int n = scn.nextInt();
if (n == 0) {
    SOP(0)
}
else {
    while (n != 0) {
        SOP(n%10);
        n = n/10
    }
}
```

1024	True	4	102
102	True	2	10
10	True	0	1
1	True	1	0
0	False ⇒	Breaks	

Digits (-6341) = Digits (6341)

-2 % 10 ⇒ -2

```
int n = scn.nextInt()
if ( n < 0 ) {
    n = n * -1;
}

if ( n == 0 ) {
    SOP(0)
}
else {
    while ( n != 0 ) {
        SOP( n%10 );
        n = n/10
    }
}
```

Print the sum of digits of a number

N = 6123 ⇒ 6+1+2+3 = 12
N = 10 ⇒ 1
N = 7 ⇒ 7

```
int n = scn.nextInt()
if ( n < 0 ) {
    n = n * -1;
```

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

Given a number. Reverse it.

N = 6123 ⇒ 3216
N = 712 ⇒ 217
N = 10 ⇒ 1
100 ⇒ 1
N = 270 ⇒ 72
N = 9 ⇒ 9
N = 0 ⇒ 0

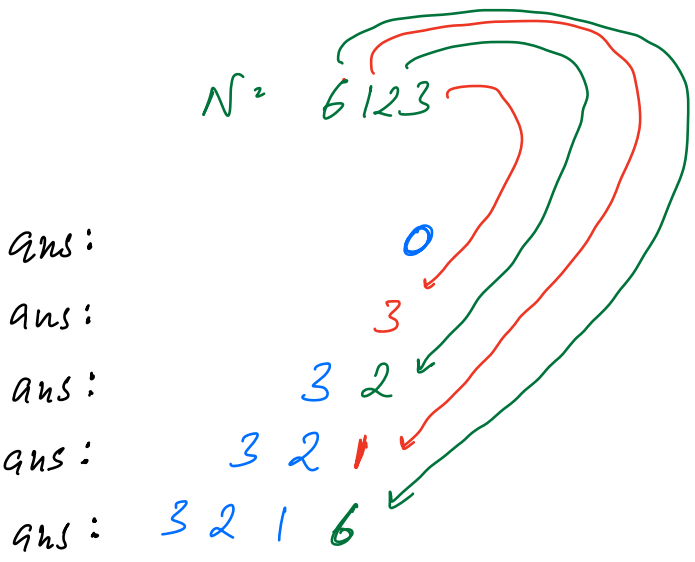
6123 Print 3216 x
int 3216

y = 314
d = 5

n = 3145

r x 10 + d ⇒ 314 x 10 + 5
⇒ 3140 + 5
⇒ 3145

r = 62
d = 1
r x 10 + d ⇒ 62 x 10 + 1
⇒ 620 + 1
⇒ 621




```
int n = scn.nextInt()
int ans = 0;
while ( n > 0 ){

    int d = n % 10;
    n = n / 10;

    ans = ans x 10 + d;
```

}

n	condition	d	n	ans
6123	T	3	612	0 x 10 + 3 = 3
612	T	2	61	3 x 10 + 2 = 32
61	T	1	6	32 x 10 + 1 = 321
6	T	6	0	321 x 10 + 6 = 3216
0	F ⇒ Break			

100	T	0	10	0 x 10 + 0 = 0
10	T	0	1	0 x 10 + 0 = 0
1	T	1	0	0 x 10 + 1 = 1
0	F ⇒ Break			

Break : 10 : 35

```
int i = 1;
while (i <= 10){
    SOP(i);
    i = i+1;
}
```

for (initialisation; condition ; update){

```
// write logic
}

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

    SOP(i);

}
```

Write code to print odd numbers till 10 using for loop

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

    if(i%2==1) {
        SOP(i)
    }

}

for(int i=1; i<=10; i=i+2) {

    SOP(i);

}
```

Factors

↳ Factor of a number N will be a number that completely divides N

24 ⇒ 1, 2, 3, 4, 6, 8, 12, 24

⇓

Factors of 24

20 ⇒ 1, 2, 4, 5, 10, 20

⇓

Factors of 20

1 ⇒ 1

Min factor N ⇒ 1

Max factor N ⇒ N

All factors of N lie in range [1, N]

```
for(int i=1; i<=N; i++){

    if(N%i==0){
        SOP(i);
    }
}
```

Prime Numbers

↳ A number which has exactly 2 factors

fact(2) ⇒ 1, 2 [Prime]

fact(11) ⇒ 1, 11 [Prime]

fact(10) ⇒ 1, 2, 5, 10 [Not prime]

fact(1) ⇒ 1 [Not prime]

Write code to check if N is prime

```
int factors = 0;
for(int i=1; i<=n; i++){
    if(n%i==0){
        factors++;
    }
}
if(factors==2){
    SOP("prime")
}
```

N = 12

i	cnt
1	1
2	2
3	3
4	4
5	4
6	5
7	5
8	5
9	5
10	5
11	5

12 6

```
int factors = 0;
for(int i=1; i<=n; i++){
    if(n%i==0){
        factors++;
    }
    if(factors>2){
        break;
    }
}
if(factors==2){
    SOP("prime")
}
else{
    SOP("composite")
}
```