

Q1: loop to print all odd numbers less than 10

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

Q2: loop for bowling all balls of an over

```
int ball = 1;
while ( ball <= 6 ) {
    SOP(ball);
    ball = ball + 1;
}
```

Q3: Print last digit of  $N$

$N \% 10 \Rightarrow$  Prints last digit

Q4: Print all digits of a number  $N$

$N = 6431$

Output

↑  
1  
3  
4  
6

$N/10 \rightarrow 6431 \% 10 \rightarrow 1$   
 $N/10 \rightarrow 643 \% 10 \rightarrow 3$   
 $N/10 \rightarrow 64 \% 10 \rightarrow 4$   
 $N/10 \rightarrow 6 \% 10 \rightarrow 6$   
 $0 \Rightarrow \text{Break}$

```

int N = scan.next Int()
while (N > 0) {
  int d = N % 10
  sop(d)
  N = N / 10
}

```

N	N > 0	d	New N
6431	T	1	643
643	T	3	64
64	T	4	6
6	T	6	0
0	F	$\Rightarrow \text{Break}$	

Input  $\Rightarrow 0$   
 Output  $\Rightarrow \text{Nothing}$

`int N = scn.nextInt();`  
`while (N >= 0) {`  $\Rightarrow$  Wrong  
 `int d = N % 10;`  
 `sop(d);`  
 `N = N / 10;`  
`}`

Infinite loop  $\Rightarrow$  Code never exits loop

`int N = scn.nextInt();`  
`if (N == 0) {`  
 `sop(0);`  
`}`  
`else {`  
 `while (N > 0) {`  
 `int d = N % 10;`  
 `sop(d);`  
 `N = N / 10;`  
 `}`  
`}`

Handling -ve numbers

digits (6431) = (-6431) digits

```

int N = sc.nextInt();
if (N == 0) {
    sop(0)
}
else {
    if (N < 0) {
        N = N * -1
    }
    while (N > 0) {
        int d = N % 10
        sop(d)
        N = N / 10
    }
}

```

Q5: Given an integer N  
Print sum of its digits

$N = 6431$   
 $Sum = 6 + 4 + 3 + 1 = 14$

$N = -321$   
 $Sum = 3 + 2 + 1 = 6$

```

int N = sc.nextInt();
int sum = 0

```

```

if ( N < 0 ) {
    N = N * -1;
}
while ( N > 0 ) {
    int d = N % 10;
    sum = sum + d;
    N = N / 10;
}
return sum;

```

Q6: Given a positive integer  $N$   
Reverse it

$$N = 6431$$

$$M = 1346$$

$$N = 320$$

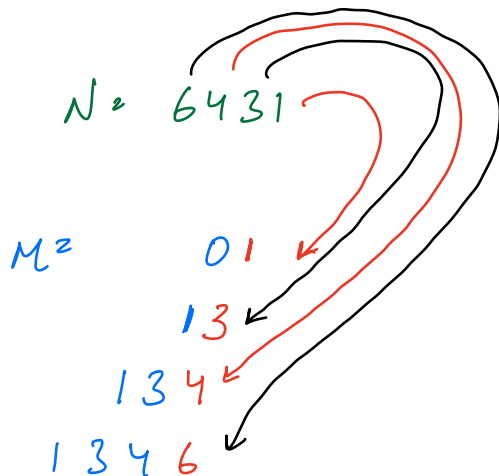
$$M = 23$$

$$N = 314$$

$$M = 3145$$

$$d = 5$$

$$N \times 10 + d = 314 \times 10 + 5 = 3140 + 5 = 3145$$



```

int N = scanf("%d", &N);
int M = 0;
while (N > 0) {
    int d = N % 10;
    M = M * 10 + d;
    N = N / 10;
}
SOP(M);

```

N	M	d	new M	new N
6431	0	1	1	643
643	1	3	13	64
64	13	4	134	6
6	134	6	1346	0
0	1346	⇒ Loop breaks		

Break : 10:20

## For Loops

// initialise

```
int i = 1
```

```
while (i <= 10) { // Condition
```

```
    SOP(i) {
```

```
    } // Logic
```

$i = i + 1$  // update  
 }

for (initialisation; condition; update) {  
     // logic  
 }

// Print 1 to N using for loop

for (int num = 1; num ≤ N; num = num + 1) {  
     SOP(num)  
 }

$i = i + 1$   
 $i++$   
 $++i$

// Print N to 1 using for loop

for (int num = N; num ≥ 1; num--) {  
     SOP(num)  
 }

Factors

Factor of N is a positive integer d

such that  $N \% d == 0$

✓

12  $\Rightarrow$  1, 2, 3, 4, 6, 12

10  $\Rightarrow$  1, 2, 5, 10

Min factor  $N \Rightarrow 1$

Max factor  $N \Rightarrow N$

Write code to print all factors of  $N$

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

```
    if (  $N \% \text{num} == 0$  ) {
```

```
        SOP(num)
```

```
    }
```

```
}
```

Prime Numbers

A number which has exactly 2 factors

factor(1) = [1] ✗

factor(11) = [1, 11] ✓

factor(0) =  $\infty$  ✗

factor(24) = [1, 2, 3, 4, 6, 8, 12, 24] ✗

Write code to check if  $N$  is prime or not



```

int cnt = 0
for(int i=1; i ≤ N; i++) {
    if (N % i == 0) {
        cnt++;
    }
}
if (cnt == 2) {
    Sol("Prime")
}
else {
    Sol("Not prime")
}

```

	N = 12		
	i	i ≤ 12	cnt
	1	✓	1
	2	✓	2
	3	✓	3
	4	✓	4
waste	5	✓	4
time	6	✓	5
	7, 8, 9, 10, 11	✓	5
	12	✓	6
	13	x	

break

```
int cnt = 0
for(int i = 1; i <= N; i++) {
    if(N % i == 0) {
        cnt++;
    }
    if(cnt > 2) {
        break;
    }
}
if(cnt == 2) {
    Sol("Prime")
}
else {
    Sol("Not prime")
}
```

1 stops the 1st  
4 parent loop

```
1 for() {
2
3 for() {
4     break;
5 }
6 }
```

4 only stop loop 3

Write code to print even till 10

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

```
    if(i%2 == 1) {  
        continue;  
    }
```

```
    sop(i)
```

```
}
```

i	i ≤ 6	output	i++
1	✓	skip	2
2	✓	2	3
3	✓	skip	4
4	✓	4	5
5	✓	skip	6
6	✓	6	7
7	✗		