Loop for bowling all balls of an over

A.

X

```
count = 1
while (count < 6) {

    //deliver a ball
    count = count + 1
}
```

count	count < 6	Task
1	true	✓ .
2	true	✓ .
3	true	✓ .
4	true	✓ .
5	true	✓ .
6	false	→ Break

B.

X

```
count = 1
while (count <= 6) {

    // deliver a ball
}
```

(deliver infinite balls)

count	count ≤ 6	Task
1	true	✓
1	true	✓
⋮		

C.

✓✓

```
    count =1
while (count <=6) {
 →  // deliver a ball
       count = count +1;

}
```

count	count<=6	Task
1	true	✓
2	true	✓
3	true	✓
4	true	✓
5	true	✓
6	true	✓
7	false →	Break

D.

X

```
    count = 0
while (count <= 6) {
        // deliver a ball

        count= count+1 ;
}
```

count →	0	1	2	3	4	5	6
	✓	✓	✓	✓	✓	✓	✓

1. initialise

2. condition

3. Task to be repeated

4. updation

```
initialise

    while (condition) {

        // task

        // updation
}
```

Ques. Given a positive integer find its last digit.

6 3 8 $\boxed{7}$

[Most significant digit] MSD ← 6

7 → LSD [Least significant digit]

N	ans
9105	5
10000	0
17543	3

last digit = n % 10

$$
\begin{array}{r l}
 & \;94 \\
10 & \overline{)\,941\,(} \\
 & \underline{90} \\
 & \;41 \\
 & \underline{\;40} \\
 & \underline{\;\;1}
\end{array}
\qquad\qquad
\begin{array}{r l}
 & \;910 \\
10 & \overline{)\,9105\,(} \\
 & \underline{90} \\
 & \;10 \\
 & \underline{\;10} \\
 & \;\;\;5 \\
 & \underline{\;\;\;0} \\
 & \underline{\;\;\;5} \;\checkmark
\end{array}
$$

$$
\begin{array}{r l}
 & \;0 \\
10 & \overline{)\,5\,(} \\
 & \underline{0} \\
 & \underline{5}
\end{array}
$$

$$1643 = 1640 + \boxed{3}$$

$$\frac{1643}{10} = \frac{(1640+3)}{10}$$

$$= \boxed{\frac{1640}{10}} + \boxed{\frac{3}{10}}$$

↳ no remainder

$\overset{(int)}{7246} / \overset{(int)}{10} \longrightarrow \boxed{724}$

724.6 ✗ $\boxed{724}$ ✓

$12345 \ \% \ 10 \longrightarrow 5$

```
        1234
10 ) 12345 (
     10
     ----
      23
      20
     ----
       34
       30
     ----
        45
        40
     ----
         5
     ----
```

Ques. Given an integer N, print all the digits from right to left

$N = \underleftarrow{6397}$

7

9

3

6

$N = \underleftarrow{420}$

0

2

4

$N = \underleftarrow{1051}$

1

5

0

1

6397 % 10 = 7

6397 % 100 = 97

6397 % 1000 = 397

6397 % 10000 = 6397

6397 % 10 → 7

6397/10

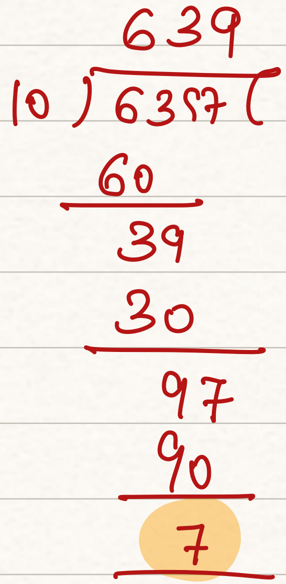
639 % 10 → 9

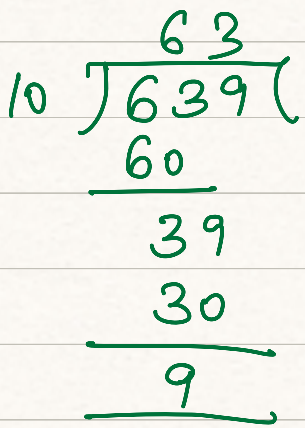
639/10

63 % 10 → 3

63/10

6 % 10 → 6





① get last digit ($n \% 10$)

② Print last digit

③ divide n by 10 ($n/10$)

```
while ( n>0 ) {
    int digit = n%10;
    S.O.Pln (digit);
    n = n /10;
}
```

Output →

2
1
9

2
1
9

n = 912

digit = 912%10 => 2

n = 912/10

n = 91

digit = 91%10 = 1

n = 91/10

n = 9

digit = 9%10 = 9

n = 9/10

n = 0

Ques. Given a integer, print the sum of all its digits.

ans

N = 623 $\quad$ 6+2+3 = 11

N = 1001 $\quad$ 1+0+0+1 = 2

N = 8142 $\quad$ 8+1+4+2 = 15

```
int N = scn.nextInt();
  int sum = 0 ;

while (N > 0) {
        int digit = N % 10;
          sum = sum + digit;
          N = N/10;

}
S.O.P (sum);
```

Sum = 0

N	N>0	digit (N%10)	sum	N=N/10
683	true	3	0+3=3	68
68	true	8	3+8=11	6
6	true	6	11+6 = $\boxed{17}$	0
0	false	→		Break

Output → 17

Ques. Given a positive integer (N), print all the perfect squares till N.

↳ whose square root is a complete integer

$\sqrt{1}$ → 1 $\quad$ $\sqrt{4}$ → 2 $\quad$ $\sqrt{9}$ → 3 $\quad$ $\sqrt{16}$ → 4 $\quad$ $\sqrt{25}$ → 5

(Input) N = 20

(Output) → 1 4 9 16

N = 50

1 4 9 16 25 36 49

N = 10

1 4 9

```
int n = 10 ;

int i = 1 ;

while (i <= n) {

    SO·P (i*i) ;

      i++;

}
```

i	i <= n	Output	i++
1	true	1	2
2	true	4	3
3	true	9	4
4	true	16	5
5	true	25	6
6	true	36	7
7	true	49	8
8	true	64	9
9	true	81	10
10	true	100	11
11	false	→	Break

Output → 1 4 9 16 25 36 49 64 81 100

```
int n = 10 ;

int i = 1 ;
                        (i)^2

while (i*i <= n) {

      S.O.P   (i*i) ;

         i++;

}
```

i	i*i <= 10	Output	i++
1	1*1 <= 10 [true]	1	2
2	2*2 <= 10 [true]	4	3
3	3*3 <= 10 [true]	9	4
4	4*4 <= 10 [false]	→	Break
	16 <= 10		

Multiple Test Cases

3 Test Cases

5 Test Cases

51
1234
490
1999
165

ans:
1
4
0
9
5

Ques. Take $\boxed{T}$ (number of test cases) as input.

for each test case take integer N as input and Print Sum of digits.

Input →

2 → T [number of test cases]
6 3 1 → Test case 1
1 5 0 2 → Test case 2

Output →

$6+3+1 \Rightarrow 10$
$1+5+0+2 \Rightarrow 8$

take input and print its sum of digits

```
int T= scn.nextInt();

while (T>0) {
    // work      → T number of times

    T--;         // decrement T by 1

}
```

T= 4

T	T > 0	Work	T--
4	true	W1 ✓	3
3	true	W2 ✓	2
2	true	W3 ✓	1
1	true	W4 ✓	0
0	false	→	Break

```
int t = scn.nextInt();

while (t>0) {

    // work → take input and print
                 sum of digits

        int n = scn.nextInt();
        int sum = 0;
        if (n<0) {
              n = n * (-1);
        }
        while (n>0) {
              int digit = n%10;
              sum = sum + digit;
              n = n/10;

}
```

```
        SO.Pm(Sum);
    t--;     //  t = t-1;
}
```

A ^B

A = 2   B = 3

$$(2)^3 \longrightarrow 8$$

$$(2)^3 \longrightarrow 2 * 2 * 2$$

$$(3)^2 \rightarrow 3 * 3$$

multiply A B times

```
int ans = 1;

int i = 1;

while (i <= B) {

        ans = ans * A;

        i++;
}
```

A = 2 $\boxed{B = 4}$

i = $\boxed{\begin{matrix}1\\2\\3\\4\end{matrix}}$ ⟶ B

Output $(2)^4$

↳ 16

i	i<=4	ans	i++
1	true	2	2
2	true	2*2	3
3	true	2*2*2	4
4	true	$\boxed{2*2*2*2}$	5
5	false	⟶	Break

ans = $\underline{2*2*2*2}$ ⇒ $(2)^4$ ⇒ 16