For Loop - 1 [Basic flow]

For Loop - 2 [New Logical Problems]

Ques. Print number from 1 to 10

```
int i = 1;             // initialization
while (i <= 10) {              // condition

      S.O.Pln(i);              // statement / task
      i++;             // updating

}
```

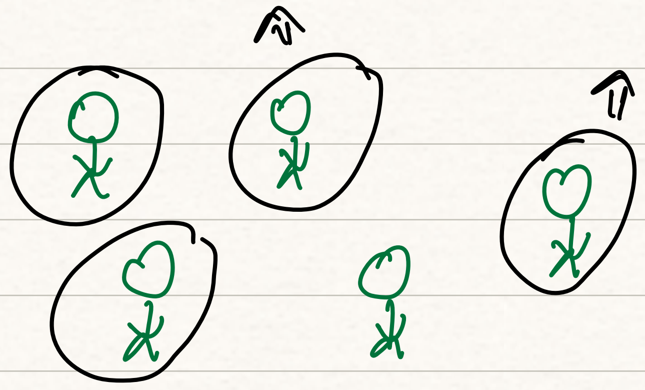
Order of execution

1. Initialisation
2. Condition check
3. Loop work
4. update

$$\begin{tikzcd} 1 \arrow[r] & 2 \arrow[r] & 3 \arrow[r] & 4 \arrow[ll, bend left] \end{tikzcd}$$

For Loops.

sum:



For every employee

1. get the salary
2. add it

Syntax →

```
                [once]
for ( initialise ; loop condition ; update ) {
                        true
        // Loop work / task to be repeated

}
```

Print numbers from 1 to 10

```
 int  i=1 ;
while  (i<=10) {
     S.O.Pln (i);
      i++;
}
```

```
for (int i=1;  i<=10 ;  i++){
          S.O.Pln(i);

}
```

Print from 1 to n.

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

               ①              ②                ④
     for( int  i=1  ;    i <= n   ;      i++){

            ③  S.O.Pln(i);

     }
```

$$\begin{tikzcd}
① \arrow[d] \\
② \arrow[d, "(true)"] \\
③ \arrow[d] \\
④ \arrow[uu, bend left]
\end{tikzcd}$$

$\boxed{n = 5}$

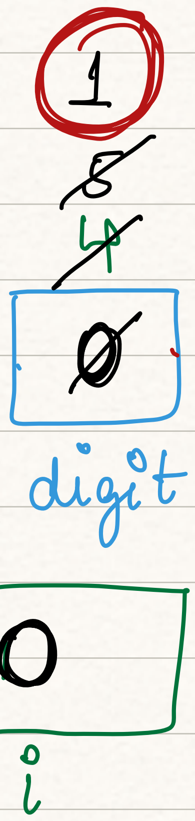
i	i <= 5	Output	i++
1	true	1	2
2	true	2	3
3	true	3	4
4	true	4	5
5	true	5	6
6	false	$\longrightarrow$	Break

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

int digit = 0;

for( int i=n ;   i>0 ;   i=i/10 ){

  → digit =   i%10;
    S.O.Pln  (digit);
}
S.O.Pln (digit);     1          4 ✓
                                5
n=     154                      1
```



i	i>0	(i%10) digit	Output	i=i/10
154	true	4	4	15
15	true	5	5	1
1	true	1	1	0
0	false	$\longrightarrow$		Break

Print the first and last digit of a number.

$n = \boxed{1}\,9\,6\,\boxed{5} \longrightarrow 1 \quad 5$

MSD (first digit) ← 1 5 → LSD (last digit)

$n = \underline{6}\,1\,0\,\underline{5} \longrightarrow 6 \quad 5$

Ques. Given a positive number N, reverse the number.

$154 \longrightarrow 451$

$N = 6123 \longrightarrow 3216$

$N = 712 \longrightarrow 217$

$N = 1000 \longrightarrow 1$

$N = 270 \longrightarrow 72$

$$\begin{tikzcd}
N = \underline{314} \arrow[dr, "314*10"'] & & \boxed{5} \arrow[ll, bend right] & \boxed{3145} \\
& \boxed{3140} \arrow[urr, "3140+5"] \arrow[urr, bend right] & &
\end{tikzcd}$$

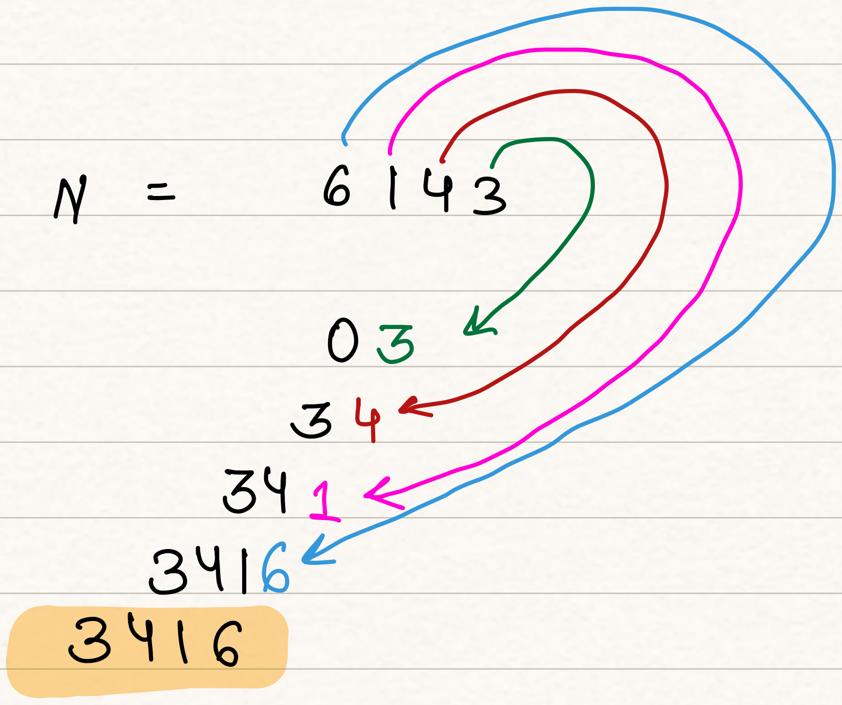
$$\boxed{314*10 + 5}$$

$$\begin{tikzcd} 617 & \boxed{1} \arrow[l, bend right] & 6171 \end{tikzcd}$$

$$\boxed{617 * 10 + 1}$$

Add the digit d to the back of the number r
"1" "617"

$$\boxed{r * 10 + d}$$



$N = 7834$

$rev = 0$

$rev = rev * 10 + (7834 \% 10)$

$rev = 0 * 10 + 4$

$rev \Rightarrow \boxed{4}$

$N = 7834 / 10 \Rightarrow 783$

$rev = rev * 10 + (783 \% 10)$

$= 4 * 10 + 3$

$= \boxed{43}$

$N = 783 / 10 = 78$

$rev = rev * 10 + (7\underline{8} \% 10)$

$= 43 * 10 + 8$

$= \boxed{438}$

$N = 78 / 10 = 7$

$rev = rev * 10 + (7 \% 10)$

$= 438 * 10 + 7$

$= 4387$

$N = 7 / 10 = 0$

1. get last digit [N % 10]
2. add to the back of rev [rev * 10 + digit]
3. N = N/10

Doubts

Bank Account -2

Balance

total no. of operation

For each operation

$\hookrightarrow$ ① type

$\hookrightarrow$ ② Amount

type = 1 (add)

type = 2 (subtract)

$\hookrightarrow$ (amount > balance)
Insufficient Funds

$\hookrightarrow$ difference

```
1000
3
1  500
2  1400
2  500
```

Balance = ~~1000~~ ~~1500~~ 100
t = 3

type = 2
amount = 500

Output →

```
1500
100
Insufficient funds
```

```
long balance = scn.nextLong();
int t = Scn.nextInt()
while (t>0) {

    int type = scn.nextInt();
    long amount = Scn.nextLong();

    if (type ==1) {
        balance = balance + amount;
        S.O.Pln( balance);

    } else {
```

```
                    if (amount > balance){
                        S.O.Pln ("Insufficient funds");

                    } else {
                        balance = balance - amount;
                    }     S.O.Pln (balance);

            }


        t--;
    }
```

Sum of Odd & Even Index digit

N = 4 5 2 4 1 2 6

7 6 5 4 3 2 1 ← Index

Odd Index → 1, 3, 5, 7

Odd Index Digit → 6 + 1 + 2 + 4 = 13

Even Idex → 2 4 6

Even Index Digits → 2 + 4 + 5 = 11

```
int   n = scn.nextInt();
 int   oddsem = 0;
  int   evensem = 0;
int   index = 1;
```

```
while (n > 0) {
        int digit = n % 10;
        if (index % 2 == 0) {
                evensum = evensum + digit;
        } else {
                oddsum = oddsum + digit;
        }
     index ++;
     n = n / 10;
}
```

Armstrong Numbers!

Sum of cubes of each digits is equal to number itself.

```
int n = Scn.nextInt();

int i = 1;

while (i <= n) {

    // check whether i is Armstrong

    int num = i;
    int sum = 0;

    while (num > 0) {

        int digit = num % 10;
```

```
                sum = sum + (digit * digit * digit);
                num = num/10;
          }
          if (sum == i ){
               S.O.Pln (i);
        }

        i++;

    }
```

count of digits

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

    int n = scn.nextInt();
     int count = 0;

    if (n == 0) {
        count = 1;
    }
     while (n>0) {
         n = n/10;
           count++;
    }
    S.O.Pln (count);

  t--;
}
```

$\boxed{n=0}$ ✓

$\hookrightarrow \boxed{1}$

count = ~~0~~ 1