Given an integer number n, return the difference between the product of its digits and the sum of its digits.

$n = 234$

$sum = 0;$
$product = 1;$

$n \% 10 = 4$

$n = n/10;$
$n = 23$

$234/10 = \quad 10)\overline{234}(23$
$\qquad \underline{20}$
$\qquad 34$
$\qquad \underline{30}$
$\qquad \boxed{4}$

$n = 23$

While $\left(\; n \underset{0}{!} = 0 \quad \right)$
$\{$

$sum = \underline{sum} + (n\%10);$

$product = product * (n \% 10);$

$n = n/10;$
$\}$

return $(product - sum);$

$\}$

Leet $\boxed{371, 1281, 250}$
done in class

$sum = 4 + 3 + 2$
$product = 4 * 3 * 2$

$n = 23/10$

$n = 2/10 \Rightarrow 0$

① Storage Classes
② Memory Layout
③ Question Session 2 — ⑦
④ Question Session 4 — ⑩

$n = 234$

$n = 234$
$rem = \; n\%10 = 4$
$n = n/10 = 23$

$rem = \; n\%10 = 3$
$n = n/10 = 2$

$rem = \; n\%10 = 2$
$n = n/10 = 0$]

```
234
int subtractProductAndSum(int n) {
    int sum = 0;
    int prod = 1;
    while ( n != 0 )
    {
        sum = sum + (n%10);
        prod = prod * (n%10);
        n = n/10;
    }
    return prod - sum;
}
```

Bitwise Operator :- Used to manipulat @ Bit level.

AND — &
OR — |
XOR — ^
Not — ~
Left — <<
Right — >>

int x = 10; → 1010 → 10
int y = 20; → 10100 → 20
int z = x & y;

$2)\overline{20}$
$2)\overline{10} \; 0$
$2)\overline{5} \; 0$
$2)\overline{2} \; 1$
$2)\overline{1} \; 0$
$\overline{0} \; 1$

0000 1010   x
000 10100   y
00000000

int z = x | y;

0000 1010
000 10100
 0 11110

int z = x^y;

0000 1010
000 10100
00 11110 => 30

XOR — x | y
x	y	
0	0	0
0	1	1
1	0	1
1	0	0

If input is same
o/p is zero
otherwise 1

0 1010

$$z = 000\cdots$$

$0 \times 2^7 \times 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

$16 + 8 + 4 + 2 + 0$

$\Rightarrow$ (80)

$z = Nx$; $000\cdots$
$1111 0101$

※ 2

Left Shift :- $z = x << 1$; [20]   ND

Discard $0000 1010$
$0 001 0100$

Right Shift :- $z = x >> 1$; [1/2]

$0000 1010$
$0 0 000 101 \rightarrow$ (5)