

③ Any base subtraction. (for  $n_2 - n_1$ )

CODE

```
boolean pos = true;
if (n1 > n2) { // if answer is negative.
    int temp = n2;
    n2 = n1;
    n1 = temp;
    pos = false;
}

int ans = 0, borrow = 0, pow10 = 1;
while (n2 > 0) {
    int res = (n2 % 10) - (n1 % 10) - borrow;
    borrow = 0;
    if (res < 0) {
        borrow = 1;
        res = res + 10;
    }
    ans = ans + res * pow10;
    pow10 = 10 * pow10;
    n1 = n1 / 10;
    n2 = n2 / 10;
}

if (pos)
    return ans;
else
    return -1 * ans;
```

④ Any base multiplication

Eg:

	2	1	5	6	$n_1$
x			7	4	$n_2$

$$\begin{array}{r}
 8624 \\
 + 15092x \\
 \hline
 159544
 \end{array}$$

Step ① extract digit of  $n2$  one by one starting from unit place

Step ② multiply with  $n1$  & find result

Step ③ add  $(result \times pow10)$  using Any Base Addition

Code :

```

int Prod (int n1, int n2, int b) {
    int ans = 0, pow10 = 1;
    while (n2 > 0) {
        int temp = n1, pow102 = 1, carry = 0, res = 0;
        while (temp > 0 || carry > 0) {
            res += ((n1 % 10) * (n2 % 10) + carry) % b;
            carry = ((n1 % 10) * (n2 % 10) + carry) / b;
            pow102 = pow102 * 10;
            temp = temp / 10;
        }
        int temp1 = ans, temp2 = res * pow10;
        carry = 0, pow102 = 1, ans = 0;
        while (temp1 > 0 || temp2 > 0 || carry > 0) {
            ans += ((temp1 % 10 + temp2 % 10 + carry) % b) * pow102;
            carry = ((temp1 % 10) + (temp2 % 10) + carry) / b;
            pow102 = pow102 * 10;
            temp1 = temp1 / 10;
            temp2 = temp2 / 10;
        }
        pow10 = pow10 * 10;
    }
}

```

}  $n2 = n2 / 10;$

} return ans;

