

1)

Counts digits a number

problem statement: Given an integer N, return the number of digits in N.

→ Input: 12345

output: 5

Explanation: The number 12345 has 5 digits.

```
while (n > 0) {
```

```
    int remainder = n % 10;
```

```
    print (remainder);
```

```
}
```

```
public static int countDigits {
```

```
    int n = 3458;
```

```
    int digits = 0;
```

```
    for (int i = 0; i < 10; i++) {
```

```
        system.out.println (
```

```
        while (n > 0) {
```

```
            n = n / 10;
```

```
            digits++;
```

```
        }
```

```
        System.out.println (digits);
```

3458  
while (3458 > 0)  
digit = 0  
n = n / 10  
n = 3458 / 10

n = 345.8  
int n = 345  
digit ++ — 1

while (345 > 0)  
n = 345 / 10

n = 34.5  
int n = 34  
digit ++ — 2

while (34 > 0)  
n = 34 / 10  
n = 3.4  
int n = 3  
digit ++ — 3

while (3 > 0)  
n = 3 / 10  
n = 0.3  
int n = 0  
digit ++ — 4

while (0 > 0)  
condition failed.



Reverse Digit of A Number

→ Input: 12345

Output: 54321

Explanation: The reverse of 12345 is 54321

```
public static int public static int ReverseNumber  
int remainder=0;
```

```
int n = 12345
```

```
while (n > 0) {
```

```
int reversed
```

```
int remainder = n % 10;
```

```
reversedNumber = reversedNumber * 10 +
```

```
n = n / 10;
```

```
}
```

```
System.out.println (reverseNumber);  
}
```

Input: 12345

```
int n = 12345
```

```
while (12345 > 0) {
```

```
int remainder = (12345 % 10) = 5
```

```
reversedNumber = 0 * 10 + 5 = 5
```

```
n = 12345 / 10 = 1234
```

```
}
```

```
5
```

```
while (1234 > 0) {
```

```
int remainder = (1234 % 10) = 4
```

```
reversedNumber = 5 * 10 + 4 = 54
```

```
n = 1234 / 10
```

$$n = 123$$

while (123 > 0) {  
 int Remainder: (123 % 10) = 3  
 reversed Number = 54 \* 10 + 3 = 543

$$n = 123 / 10$$

$$n = 12$$

while (12 > 0) {  
 int Remainder: (12 % 10) = 2  
 reversed Number = 543 \* 10 + 2 = 5432

$$n = 12 / 10$$

$$n = 1$$

while (1 > 0) {

int Remainder: (1 % 10) = 1  
 Reversed Number: 5432 \* 10 + 1 = 54321

$$n = 1 / 10$$

$$n = 0.1$$

while (0 > 0)

Condition failed.

3) Palindrome Number

Input: 4554

Output: Palindrome Number

Explanation: The reverse of 4554 is 4554 and therefore it is palindrome



if (int n == Reversed Number) {

s.o.p ("palindrome Number");  
}

else {

s.o.p ("Not palindrome Number");  
}

N = 7789

int n = 7789;

~~Rev~~ public static void palindrom {

int reversed Number = 0;

while (7789 > 0) {

int Remainder r = 7789 % 10; = 9

Reversed Number = 0 \* 10 + 9 = 9

n = 7789 / 10 = 778

}

while (778 > 0) {

778 % 10 = 8

9 \* 10 + 8 = 98

n = 778 / 10

= 77

98

```

while ( 7770 ) {
    77 % 10 = 7
    987 * 10 + 7 = 987
    n = 77 / 10
    = 7.7 = 7

```

```

while ( 7 > 0 ) {
    7 % 10 = 7
    987 * 10 + 7 = 9877
    n = 7 / 10
    = 0.7 = 0

```

```

while ( 0 > 0 ) {
    Condition failed.

```

if ( 7789  $\neq$  9877 )

Not palindromic Number.

Print all divisors of given Number  
Example:

Input = 36

Output = [ 1, 2, 3, 4, 6, 9, 12, 18, 36 ]

Explanation: The divisors of 36 are 1, 2, 3, 4, 6, 9, 12, 18, 36.

int N = 36

~~for ( i = 1; i <= N; i++ ) {~~

for ( i = 1; i < N; i++ ) {  
 if ( N % i == 0 ) {  
 print (i)

}



3) Find GCD of two Number

example 1:

Input:  $N_1: 9$   $N_2: 12$

output:

factor of 9 is 1, 3, 9

factor of 12 is 1, 2, 3, 4, 6, 12

common factor is 1, 3

GCD :- 3

Here

```
for (int i = 1; i <= 9; i++) {
```

```
    if (9 % i == 0 && 12 % i == 0) {
```

```
        Gcd = i;
```

4) check if a Number is Armstrong Number or not

Example 1:

Input:  $N = 153$

output: true

Explanation:  $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$

```
Main () {
```

```
    int n = 153;
```

```
    int digits = 0;
```

```
    int sum = 0;
```

```
    while (n > 0) {
```

```
        digit++;
```

```
        temp = n / 10;    temp = 153 / 10 = 15
```

```
        n = temp;
```

```
    }
```

```

temp = n;
while (temp > 0) {
    int digit = temp / 10;
    int power = 1;
    for (int i = 0; i < digits; i++) {
        power = digit * power;
    }
    sum += digit;
    temp /= 10;
}
return sum;
}

```

153  
1570  
153%10 = 3  
0+5  
temp

Prime Number

main () {

int n = 2

if (n <= 1) {

return boolean prime (int n)

if (n <= 1) {

return false;

}

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

if (n % i == 0) {

return false;

}

}

return true;

}