

Basic Maths Assignment Question

Date : / /

Page No.

1. count digits in number

A.S. #

Given an integer N , return number of digits in N .

Ex. 1

INPUT : $N = 12345$

output : 5.

$$12345 \div 10 = 1234 \dots 5$$

$$12345 \div 10 = 1234 \dots 5 \quad (\text{int} = 1234)$$

$$1234 \div 10 = 123 \dots 4$$

$$1234 \div 10 = 123.4 \quad (\text{int} = 123)$$

public static void main {

 int n = 12345; 123 / 10 = 12

 int digit = 0; 12 / 10 = 1.2 = 1

 while (n > 0) {

 n = n / 10; 1 / 10 = 0.1 = 0

 digit++;

 } // End of while loop
 System.out.println("Number of digits = " + digit);
}

2. $N = 7789$

output : 4

$$7789 \div 10 = 778 \dots 9$$

$$7789 \div 10 = 778.9$$

public static void main {

 int n = 7789;

 int digit = 0;

 while (n > 0) {

 n = n / 10;

 digit++;

while ($7789 > 0$) {

① int $7789 = 7789 / 10$

778.9 digit ++;

9

s.o.p(0); t = 778

while ($778 > 0$) {

$778 = 778 / 10$

77.8 digits ++;

② s.o.p(0); t = 77

while ($77 > 0$) {

$77 = 77 / 10$

③ digits ++;

7 digits = 1

s.o.p(0); t = 7

while ($7 > 0$) {

④ $7 = 7 / 10$

0 digits ++;

3

s.o.p(0); t = 0

output = 4

2. check if number is palindrome or not

PS: give an integer N, return true if N is
palindrome else return false.

for (int i=0; i<n; i++) {

For (int j=n-1; j>0; j--) {

public static boolean (int N) {

int original Number = N;

int reverse Number = 0;

while (N != 0) {

int last digit = N % 10;

reverse Number = reverse Number * 10 + last

digit + N/10;

Ex: 4554 → 4554 % 10 = 4

4554 / 10 = 455

→ public static boolean (4554)

int original Number = 4554;

int reverse Number = 0;

while (4554 != 0) {

int last digit = 4554 % 10.

original Number = reverse Number,

(No. is Palindrome) print(()); = ~~4554~~ 4554

else print(No. is Not Palindrome);

public static boolean (4554)

int original Number = 0;

4.5 while ($4554 \neq 0$) {
 If $\{$ int last digit = $455 \mod 10$
 original Number = reverse Number;
 print(); (No. is Palindrome)
 else $\{$ print (No. is Not Palindrome)
 public static boolean (4554)
 int original Number = 0;

4.5 while ($4554 \neq 0$) {
 If $\{$ int last digit = $455 \mod 10$
 original Number = reverse Number;
 print();

public static boolean (4554)
 int original Number = 0;
 while ($4554 \neq 0$) {
 If $\{$ int last digit = $455 \mod 10$
 original Number = reverse Number;
 print(); (No is Palindrome)
 else: print (No. is Not Palindrome)
 output: 4554

Q] AN = 7789
 Output : Not Palindrome

public static boolean (4554)(7789)
 int original Number = 0;
 while ($4554 \neq 0$) {
 If $\{$ int last digit = $7789 \mod 10$
 original Number = reverse Number;
 print(); (No is Palindrome)

Output :-

else {

print C Number is Not palindrome.

}

Ex. Find GCD of two Numbers.

PS :-

Given two integr N1 & N2 find their greatest common divisor.

Ex. 1.

Input N1 = 9 N2 = 12

→ public static void main() {

```
int n1n = math.gcd(n1, n2);  
int gcd = 1;  
for (int i = n2n; i > 1; i--) {  
    if (n1 % i == 0 && n2 % i == 0)  
        gcd = i;  
}
```

return i;

if (n1 == 0) {

return n2;

3

else {

return n2;

8

if ($g = 0$) {

 return 12;

 3

 else {

 return g;

 3

 3

factors of 9 = 1, 3, 9

factors of 12 = 1, 2, 3, 4, 6, 12

common factors = 1, 3, out of which 3 is greatest hence it is GCD

Ex. 2.

$n = 20, n_2 = 15$

public static void main () {

 int n, n₂ = math.n2.n (n, n₂);

 int gcd = 1

 for int i = n, n₂; i > 1; i--) ;

 if (n.i-1 * n₂.i-1 == 1);

 2

 return 1;

 3

 3

 if (20 == 0) {

 return 15;

 8

else

return 15;

5

5

factors of 20 = 1, 2, 4, 5

factors of 15 = 1, 3, 5

common factors : 1, 5 out of which 5
is greatest hence it is GCD.

Q. Q. check point all divisors of given Number.

PS : Given integer N, return all divisors of N.

Ex. 1. N = 86

→ Output should be 1, 2, 43, 86

public static void factors (int n) {

for (int i=1; i<=Math.sqrt(n); i++)

if (n % i == 0) {

System.out.print (n / i);

if (n / i != i) {

System.out.print (n / i);

8

8

8

8

1) 36

Date / / /
Page No. / /

1. $\text{for } i=1; i \leq \text{Math.sqrt}(n); i++)$
 $\{\text{s.o.p.}(i^2);$
 $\}$

2. $\text{for } i=1; i \leq \text{Math.sqrt}(n); i++)$
 $\{\text{s.o.p.}(i^2);$
 $\}$

3. $\text{for } i=1; i \leq \text{Math.sqrt}(n); i++)$
 $\{\text{s.o.p.}(i^2);$
 $\}$

4. $\text{if }(i^2 == n)$
 $\{\text{s.o.p.}(i^2);$
 $\}$

5. $\text{if }(i^2 < n)$
 $\{\text{s.o.p.}(i^2);$
 $\}$

6. $\text{if }(i^2 > n)$
 $\{\text{s.o.p.}(i^2);$
 $\}$

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

S

IF ($i^2 - 1 \leq n \leq i^2$) \Rightarrow

$\Rightarrow O.P(\ln(n) \cdot (i^2 - i)) \approx$

3

3

3

Output = 1, 2, 3, 4, 6, 12.

Q.6. Check if No. is prime or not.

Ex-1. $N = 2$

Output = True.

```
public static boolean isPrime(int n) {
    int count = 0;
    for (int i=1; i <= Math.sqrt(n); i++) {
        if (n % i == 0) {
            count++;
        }
    }
}
```

count++;

if ($n/i \neq i$) {

count++;

3

3

3

IF ($count == 2$) {

return true;

3

return false;

3

2 is prime No., because it has two
divisors 1 and 2. (itself).

Q)

N = 10

OUTPUT: false



```
public static boolean isPrime(int n) {
    int count = 0;
```

```
    for (int i = 2; i < Math.sqrt(n);)
        count++;
```

```
    if (n / i != 1)
        count++;
```

```
    count++;
```

3

3 numbers which are divisible by 10

3, 10, 20, 30, 40, 50, 60, 70, 80, 90

if (count == 2)

return true;

3

else
 return false;

3

but it has 4 divisor, 1, 2,

5 and 10 so it is

not prime no.

condition is false.