(#) Whether a number is prime or not.

$$\boxed{10} \Rightarrow \frac{10}{2}, \frac{10}{3}, \frac{10}{4}, \frac{10}{5}$$

n

[i runs from 2 to n/2]

if (n % i == 0)

if (10 % 2 == 0) ⇒ Not a Prime number

for example: n = 7   i = 2 to 3

(7 % 2 == 0) ⇒ false ⎫ Prime
(7 % 3 == 0) ⇒ false ⎭ Number

{ log = 0 ?   n = 7   i = 2 ✓
                           i = 3 ✓

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

if (n % i == 0)   ⇒ 7 % 2
                         7 % 3

```
        {
            flag = 1;        n = 10
            break;
        }
    }                  10 % 2
}                        = 0

if (flag == 0)
    printf ("prime");

else
    printf ("not prime");
```

(#) No of digits in a integer number.

$n = 1234$ count = 0

$n = n/10 = 1234/10 = 123$ count = 1

$n = 123/10 = 12$ count = 2

$n = 12/10 = 1$ count = 3

$n = 1/10 = 0$ count = 4

while (n % = 0) ← not equal to
{
  $
  ?
}

$n = 512 = 5+1+2 = 8$ Sum = sum+rem

$rem = n \% 10 = 512 \% 10 = \boxed{2}$

$n = n/10 = 512/10 = 5)$

$rem = 51 \% 10 = \boxed{1}$ Sum = sum + rem.

$n = n/10 = 51/10 = 5$

$rem = 5 \% 10 = \boxed{5}$

$n = 5/10 = 0$