Quiz 1

```
n = 1
while ( n ≤ 10) {
        SoP( n);
      n =   n+2;
}
```

$1 \xrightarrow{+2} 3 \xrightarrow{+2} 5 \xrightarrow{+2} 7 \xrightarrow{+2} 9$

Quiz 2

```
 count = 1;
while (   count ≤ 6) {
      // deliver  a   ball
      count = count+1;
}
```

1 → N

1 2 3 4 5 6

0 1 2 3 4 5

0 → N-1

Quiz 3

```
  i = 1;
while(  i ≤ 10) {
        SoP( i );
        i = i+1;
}
```

1 → 10

Q1   Given a   number, print its last digit n % 10

N = 4358 ⇒ N % 10 ⇒ 8

Q2 Print all digits of N in new line

N = 6341

Output

1

4

3

6

N/10 6341 % 10 → 1

N/10 634 % 10 → 4

N/10 63 % 10 → 3

N/10 6 % 10 → 6

0 ⇒ Break / Stop the loop

```
while ( n > 0 ){
        SOP( n %10 )
        n = n/10;
}
```

N	N > 0	N % 10	N = N/10
6341	T	1	6341/10 = 634
634	T	4	634/10 = 63

63	T	3	$63/10 = 6$
6	T	6	$6/10 = 0$
0	F	⇒ Loop breaks	

Output for N = 0 ⇒ 0

```
if ( n == 0 ){
        SOP( 0 );
}
else {
    while ( n > 0 ) {
            SOP( n % 10 )
            n = n/10;
    }
}
```

n ≥ 0 → Infinite loop

What if N < 0

```
if ( n == 0 ){
        SOP( 0 );
}
else {
    if( n < 0 ){
        n = n * -1;
    }
    while ( n > 0 ) {
```

dig(−6341) =	dig(6341)
1	1
4	4
3	3
6	6

−6341 * −1

⇓

```
        SoP( n %10)
        n =  n/10;
    }
}
```

6341

Q3 Print sum of digits of N
⇓
Non - negative

N= 6341

Sum= 14

```
    int sum= 0;
  while( n > 0 ){
         sum =  sum + n% 10;
         n =  n/10;
  }
  SoP( sum);
```

Q4 Given N reverse it

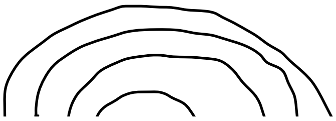
N= 6341   Rev= 1436

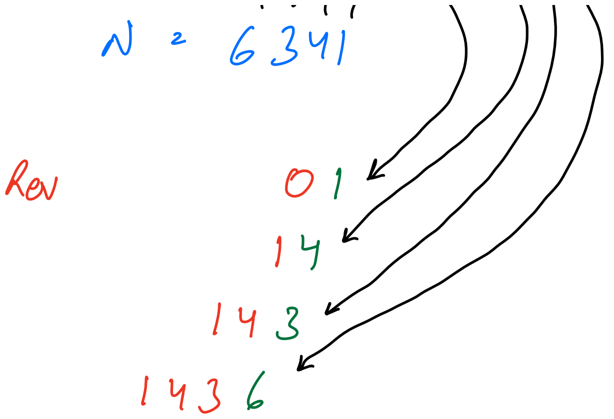
N= 634   (N)(d) ⇒ 6341

d= 1   10×N+d ⇒ 634×10+1

⇒ 6340+1

⇒ 6341





```
rev = 0
while ( n > 0 ) {
        d = n % 10
        rev =   rev x10 +d
        n =   n/10
}
```

rev	n	n>0	d	rev	n
0	6341	T	1	1	634
1	634	T	4	14	63
14	63	T	3	143	6
143	6	T	6	1436	0
1436	0	F	⇒ Loop	breaks	

Break – 10:15

For loop

```
initialisation
while ( condition ){
        do   work
        update
}
```

```
for ( initialisation ; condition; update ){

                do   work
}
```

```
for ( int i = 1; i ≤ 10; i = i+1 ){
                SOP ( i );
}
```

Q5 Print all odd numbers from 1 to N

```
for( int i=1; i≤ N; i = i +2 ){
                SOP( i );
}
```

Factors

A factor x of N is a positive

integer such that N % x == 0
N is a multiple of x

12 ⇒ 1, 2, 3, 4, 6, 12
20 ⇒ 1, 2, 4, 5, 10, 20

Min factor ⇒ 1
Max factor ⇒ N

All factors of N lie in the range
1 to N

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

        if( N% i == 0 ){
              SOP( i );
        }
}
```

Prime Numbers

A number which has exactly 2 factors

12 ⇒ 1 2 3 4 6 12 ✗
10 ⇒ 1 2 5 10 ✗
5 ⇒ 1 5 ✓

37 ⇒ 1 37 ✓

Q6 Given N check if it is prime or not

```
int cnt = 0;
for (int i = 1; i ≤ N; i++ ){

        if (N% i == 0){
              cnt++;
        }
}
if ( cnt == 2){
        SOP( Prime);
} else {
        SOP( Not prime);
}
```

Break

12

1

2

3 ⇒ We know count is 3

4 Hence, 12 is not prime

6

12

```
int  cnt=0;
for (int  i=1; i≤N; i++ ){

        if (N% i == 0){
              cnt++;
        }
        if (cnt ≥ 3){
              break;          // Intentionally
                                 break loop
                                 befor condition
        }
}
if ( cnt == 2){
        SOP( Prime);
} else {
        SOP( Not prime);
}
```

T- testcases