

## ☐ Level 1 – Basic Loop Logic

1. Sum of 10 numbers
  2. Find factorial of a given number ( $n!$ )
  3. Print sum series  $1 + 4 + 9 + 16 + \dots + n$
  4. Print digits of a number in reverse order
  5. Print sum of digits of a given number
  6. Perform multiplication without using  $*$  operator
  7. Calculate  $x^y$  without using power operator
- 

## ☐ Level 2 – Loops with Conditional Logic (if inside loop)

8. Print numbers divisible by 2 between two numbers
9. Find factors of a given number
10. Find the sum of all divisors of a number
11. Find the GCD (Greatest Common Divisor) of two numbers
12. Find the LCM (Least Common Multiple) of two numbers

13. Check whether a number is a Perfect Number.

(A perfect number is a positive integer that is equal to the sum of its positive divisors, excluding the number itself. A divisor of an integer  $x$  is an integer that can divide  $x$  evenly.)

14. Check whether a number is a Harshad Number.

(An integer num which is divisible by the sum of its digits is said to be a Harshad number.)

15. Check whether a number is Prime or Not

16. Print the Fibonacci Series

17. Find the sum of series  $1 - 2 + 3 - 4 + 5 - 6 + \dots \pm n$ ,

Check if given number is a Happy Number or not

Check if given number is a Ugly Number or not.

(a n ugly number is a positive integer which does not have a prime factor other than 2, 3, and 5.)

1. Find the sum of  $1+(1+2)+(1+2+3)+\dots(1+2+3+\dots+n)$

2. Power without using multiplication and power operator

3. implement a pattern

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

4. implement a pattern

\*

\* \* \*

\* \* \* \* \*

5. implement a pattern

1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

6. Find a Largest Element in array

7. Find a Smallest Element in array

8. Search for an element in Array

9. Reverse an Array Elements

27.

```
*  
* *  
* * *  
* * * *  
* * * * *
```

28.

```
1  
1 2  
1 2 3  
1 2 3 4  
1 2 3 4 5
```

29.

```
A  
B B  
C C C  
D D D D  
E E E E E
```

30.

```
* * * * *  
* * * *  
* * *  
* *  
*
```

31.

```
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

32.

```
      *
    * * *
  * * * * *
* * * * * * *
* * * * * * * *
```

33.

```
      1
    2 3 2
  3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
```

34.

```
* * * * * * * * *
 * * * * * * *
  * * * * *
   * * *
    *
```

35.

```
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
```

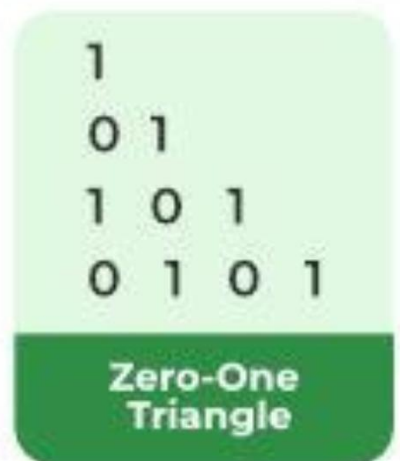
36.

```
1
2 3
4 5 6
7 8 9 10
```

37.



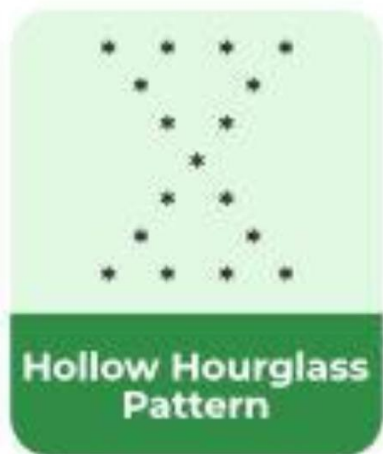
38.



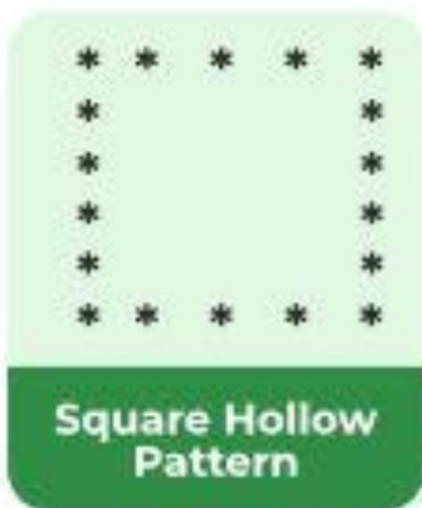
39.



40.

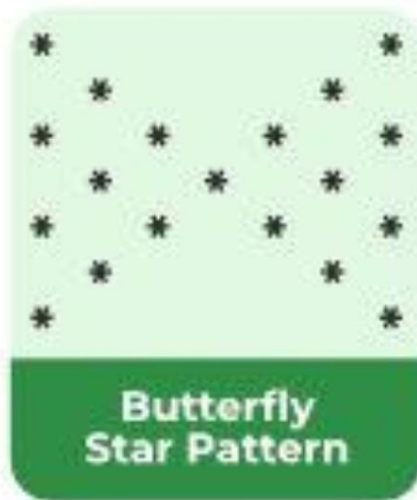


41.





42.



43.

