

1) Even / odd

1. Start
2. Print input ~~as~~ (int) number
3. Read input
4. if number $\% 2 == 0$
Print even number
5. else print odd number
6. End

2) factorial

1. Start
2. Read number (num)
3. Set $i=1$, fact = 1
4. Use for loop till i equals num. $for(i=1; i \leq num; i++)$ fact = fact * i
5. Print fact
6. End

3) Factorial using recursion

1. Start
2. Read number (num)
3. Call function fact (num)
4. Print fact
5. fact (num)
 1. if $num == 1$ then
return 1
 - else fact = fact * fact (num)
return fact
6. End

4. Swap 2 numbers without using 3rd variable

1. start
2. Read num1, num2
3. num1 = num1 + num2
- num2 = num1 - num2
- num1 = num1 - num2
4. Print num1, num2
5. Stop.

5. How to check positive or negative number.

1. start
2. Read number (num)
3. if num > 0
 Print 'positive number'
else if num < 0
 Print 'negative number'
else Print 'Zero number'
4. Stop

6. leap year

1. start
2. Read year
3. if Year % 4 == 0 && Year % 100 != 0
 OR
 Year % 400 == 0
 Print 'leap year'
4. else print 'not a leap year'
5. Stop

7. Printing 1 to 10 without using loop

1. Start
2. Define method printwithoutloop which takes int num
3. If num <= 10 print n
call printwithoutloop(n+1)
4. If n > 10, stop recursion and return
5. In the main method, call printwithoutloop = 1 ~~initially~~ initially.

8. Printing digits of a number.

1. Start
2. Read integer from user.
3. Use a while loop until 0

```
while (num != 0)  
    int digit = num % 10  
    Print digit  
    num /= 10
```
4. Stop

9. Factors of a number.

1. Start
2. Read number (num)
3. Use for loop int i=1 ; i<= num ; i++)
4. Use if loop in for

```
if      num % i == 0
```
5. Print i
6. Stop

10. Sum of digits of a number.

~~Read number and initialise it to zero~~

1. Start

2. Read number.

3. Print function total (num)

4. total (int n) is

 while ($n \neq 0$) {

 int sum = sum + (n % 10);

$n = n / 10$

 Return sum.

5. Stop

11. Smallest of 3 numbers (a, b, c)

1. Start

2. Read the numbers

3. if $a < b$ && $a < c$

 Print "a is smallest"

4. else if $b < c$

 Print "b is smallest"

5. else

 Print "c is smallest"

12. Add 2 numbers without arithmetic operators

1. Start

2. Read num1, num2

3. Use for loop for $i=1, n \leq num2, i++$
 $num1++$;

4. Print num1

5. Stop

13. Reverse of a number

1. start
2. Read
3. ~~A declare reverse = 0~~
4. ~~while number~~

1. Start
2. Read number num
3. declare reverse = 0
4. while number != 0
 remainder = num % 10
 reverse = reverse * 10 + remainder
 number = number / 10
5. Print reverse
6. End.

14. GCD of @ numbers.

1. Start
2. Read num1, num2
3. declare gcd = 1
4. initiate for loop
 for (int i=1; i <= num1 && i <= num2, i++)
 4.1 if (num1 % i == 0 && num2 % i == 0)
 gcd = i
5. Print gcd
6. Stop

15. LCM of two numbers

1. Start
2. Read num1, num2.
3. if num1 > num2
 max = increase = num1
 else max = increase = num2.
4. while (num1 != 0 & & num2 != 0) {
 if max % num1 == 0 & & max % num2 == 0
 lcm = max;
 break;
 }
 max = max + increase;
5. Print lcm
6. Stop

16. Lcm of 2 numbers using prime factor method

1. Start
2. Read num1, num2.
3. Calculate minimum of step 2. (min)
4. Initialise lcm = 1.
5. for i=2; i ≤ min/2+1 to i++)
 while (n1 % i == 0 & & n2 % i == 0)
 lcm = lcm * i;
 num1 = num1 / i;
 num2 = num2 / i;
6. end while
7. end for
8. lcm = lcm * n1 * n2
9. Print lcm
10. end

17. Palindrome

1. start
2. Read num
3. declare sum = 0, temp
4. while (temp > 0)
 int reverse = temp % 10
 sum = (sum * 10) + reverse
 temp = temp / 10
5. if (num == sum)
 Palindrome
6. else not a palindrome
7. stop.

18. Print prime factors of a number

1. start
2. Read num
3. declare c = 2
4. while num > 1
 if num % c == 0
 print c
 num = num / c
 else
 c++
10. end while
11. stop

19. Print even number series .

1. start
 2. Read number (num)
 3. for ($i = 0$, $i \leq num$, $i++$)
 $\quad \quad \quad \underline{i = i}$ print i
 $\quad \quad \quad i = i + 2$
 4. end
-

20. Print odd number series

1. start
 2. Read num
 3. for ($i = 1$, $i \leq num$, $i++$)
 print i
 $i = i + 2$
 4. end
-