I/O + Operation:

1. Take two numbers as input: length and width of a rectangle and print the area of that.
2. Take a number as input: radius of a sphere and print the volume of that.
3. Take two numbers as input: radius and height of a cylinder and print the volume of that.
4. Take two numbers as input: radius and height of a cone and print the volume of that.

Condition:

5. Take a number N as input and print whether it is even or odd.
6. Take three numbers as input: the three angles of a triangle and whether the triangle is valid or not (angle value should be such that 0 < value < 180).
   [*Recall that a triangle is valid if the sum of all the three angles is equal to 180 degrees*.]
7. Take two integers: x and y as input and print “x is divisible by y” as output if that holds. Otherwise, print “x is not divisible by y”.
   [*Recall that you can use the % operator to get the remainder when one number is divided by another. In particular, n%d evaluates to the remainder when n is divided by d. For example, 8%3 is 2 because if you divide 8 by 3, you get a remainder of 2*.]
8. Write a C program that takes as input two positive integers n and d and outputs whether d is a proper divisor of n.
   [*Recall that a proper divisor of a positive integer n is a divisor of n that is not equal to n. So, 1, 2, and 3 are proper divisors of 6 but 6 isn’t.*]
9. Take an integer as input for a year and check whether it is a leap year or not.

Looping:

10. Take an integer N as input and find its factorial (N!).
11. Write a C program to print the nth Fibonacci number.
12. Take an integer N as input and check whether N is a prime number or not.
    [*A natural number (1, 2, 3, 4, 5, 6 etc.) is called a prime number if it is greater than 1 and has only two factors: 1 and the number itself.*]
13. Write a C program that takes as input a positive integer n and outputs its largest proper divisor.
14. Write a C program that takes as input a positive integer n and outputs the number of proper divisors of n.
15. A positive integer n is called *Nice* if it has at least three proper divisors and is equal to the sum of its three largest proper divisors. For example, 6 is *Nice* because its

three largest divisors (in descending order) are 3, 2, and 1 and 6=3+2+1. Write a C program that takes as input a positive integer n and outputs whether or not it is *Nice*.

16. Take an integer N as input and check whether N is a power of 2.
17. Take an integer N as input and print the largest power of 2 that is also a divisor of N.
18. Take an integer N as input and print the largest power of 2 that properly divides N. [*Recall that a proper divisor of a positive integer n is a divisor of n that is not equal to n. So, 1, 2, and 3 are proper divisors of 6 but 6 isn't.*]
19. Take two numbers as input: x and y. Now, print n for which the following relationship holds:

$$x = y^n$$

Otherwise, print "could not find n". Assume that n is a natural number and $2 < n < 15$.

20. Take an integer N as input and print the number of digits in N. [*e.g. N=2359 should output 4, N=900 should output 3 etc.*]
21. Take a number N as input and print how many even and odd digits it has.
22. Take an integer N as input, reverse it and then print that. [*e.g. N=2359 should output 9532, N=900 should output 9 etc.*]
23. Take a number N as input and print whether it is a palindrome or not. [*A palindromic number is a number that remains the same when its digits are reversed. For example, 984489 and 12321 are palindromes, however, 1234 and 985489 are not.*]
24. Take a number N as input and print whether it is an *Armstrong* number or not. [*An Armstrong number is an n-digit number that is equal to the sum of the nth powers of its digits. For example, 407 is an Armstrong number as*

$$407 = 4^3 + 0^3 + 7^3 = 64 + 0 + 343 = 407$$*]*

25. Take an integer N as input, swap the last and second last digit of it and then print that.
26. Take an integer N having an even number of digits as input, split the first and second half as separate numbers. [*For example, split 984489 into 984 and 489*]
27. Take an integer N having an even number of digits as input, set zero to the odd positions of that integer and output that. [*For example, 984489 should produce 904080 and 1234 should produce 1030*]

Nested Looping:

28. Take an integer N as input and print all prime numbers from 2 to N inclusive.
29. Print the following pattern for input n.

n=3	*** ** *
n=5	***** **** *** ** *

30. Print the following pattern for input n.

n=3	* *** *****
n=5	* *** ***** ******* *********