1. WAP to check whether a number is positive. If Positive, print a message Positive.
2. WAP to display "Hello World" if the number is greater than 1 and lesser than 5.
3. WAP to check whether the given number is divisible by 3 or not. If divisible, print "Fizz".
4. WAP to check whether a given number is divisible by 2 and 6. If satisfied, convert the given number into a complex number.
5. WAP to check whether a given input is divisible by 3 or 5. If satisfied, convert the number into a list.

   i/p: 30

   o/p: [‘3’, ’0’]
6. WAP to check whether a given number is a multiple of 5 or not.
7. WAP to check whether the given input is 0 or not. If 0, print 0.
8. WAP to check whether a number is negative. If negative, print a message Negative.
9. WAP to check whether a number is even or not. If even, store the value inside a list.
10. WAP to check whether a number is odd or not. If odd, store the value inside a tuple.
11. WAP to check if a given number is even or not. If yes, print the power of 4 of the given number.
12. WAP to take input from the user as a number and check whether it is numeric or not. If yes, take the number, add some value, and print it.
13. WAP to check whether a given value is divisible by 5 and 7. If divisible, display the square of the value.
14. WAP to check whether the last digit of a given value is greater than 5 or not. If greater, perform the bitwise right shift operator (shift by 2).
15. WAP to check whether a given value is divisible by 3 and less than 30. If satisfied, display the square of the value.

16. WAP to check whether a given value is even and divisible by 4. If satisfied, display the cube of the value.

17. WAP to check whether a given value is even or not. If even, store the value inside a list.

18. WAP to check whether a given value is negative or even. If satisfied, display the last digit of the value.

19. WAP to check whether a given value is negative or odd and divisible by 4. If satisfied, display the cube of the value.

20. WAP to check whether a given ASCII value is divisible by 4 and even. If satisfied, display the ASCII character.

21. WAP to check whether a given value is between 45 and 125, divisible by 4 and 5, and even. If satisfied, display the ASCII character.

22. WAP to check whether a given value is between 25 and 100 and divisible by 4 and 5. If satisfied, display multiplication of the value with 5.

23. WAP to check whether a given number is an integer and odd. If satisfied, check if it is divisible by 5 and display the result.

24. WAP to check whether a given value is an integer or not. If integer, convert the value to string and display it.

25. WAP to check whether a given value is less than 125 and greater than 60. If satisfied, take a name and extract the middle character, then display it.

26. WAP to check whether two given integers are equal or not. If equal, perform addition and display the result.

27. WAP to check whether two values are equal or not. If equal, perform multiplication of the two numbers, divide the result by 3, and display it.

28. WAP to check whether a character is an alphabet or not. If alphabet, store it inside a dictionary with character as key and ASCII value as value.

29. WAP to check whether a character is uppercase or not. If uppercase, store it inside a dictionary with character as key and ASCII value as value.

30. WAP to check whether a character is uppercase or not. If uppercase, convert to lowercase and store inside a dictionary (key=character, value=ASCII value).

31. WAP to check whether a character is lowercase or not. If lowercase, perform replication operation on the character.

32. WAP to check whether a character is a digit (ASCII number) or not. If yes, convert into integer.

33. WAP to check whether a character is a special symbol or not. If yes, display the symbol with ASCII value.

34. WAP to check whether a character is an even number or not. If even, display it.

35. WAP to check whether two integers a and b satisfy: any integer is 10 or their sum is 10. If yes, perform sum and display the square of the result.

36. WAP to check whether a character is a vowel or not. If vowel, print the next character.

37. WAP to check whether a character is not a vowel. If not vowel, print the previous character.

38. WAP to check whether a given value is a single value data type or not. If yes, display it as a single value data type.

39. WAP to check whether a given value is a multi-value data type or not. If yes, store it inside a tuple and display it.

40. WAP to check whether a given value is an immutable data type or not. If yes, store it inside a set and display it.

41. WAP to check whether the middle element of a list is odd or not. If odd, print "Odd" and the number, else print the number directly.

42. WAP to return a new string with "not" prefixed to the front of the original string. Return the string unmodified if it already begins with "not".

43. WAP to check whether two variables point to the same memory location or not. If yes, print the address of both variables.

44. WAP to check whether the length of a string is even or not. If even, display the sequence of characters from index 0 to length-2.

45. WAP to check whether two strings have the same values or not. If not same, swap the values and display them.

46. WAP to check whether a given character is a special symbol or not. If yes, store it as key and ASCII value as value in a dictionary.

47. WAP to check whether a given key is present in a dictionary or not. If not present, append the new key inside the dictionary.

48. WAP to display a new string with the first and last characters of a given string switched.

49. WAP to check whether the last digit of a number is 2. If yes, print the last digit.

50. WAP to check whether two integers a and b satisfy: any integer is 10 or their sum is 10. If yes, perform sum and display the square of the result.

51. WAP to check whether a given string (str) and one integer (n) value produces a new string devoid of the character at index n. Assume n is valid.

52. WAP to check whether two strings are equal or not. If not equal, print both strings along with their lengths. If equal, ignore.