1. Simple print statements
    - Write a program to print a message, such as ‘Hello IIIT Delhi’.
    - Now change that program to print two messages, such as ‘Message from ME’, and ‘Hello IIIT Delhi’ **on the same line**.
    - AGAIN change that program to print two messages, such as ‘Message from ME’, and ‘Hello IIIT Delhi’ **on two different lines**.
    - Again print **only one statement** so that the two messages, such as ‘Message from ME’, and ‘Hello IIIT Delhi’, are printed on two different lines.
    - Lastly, modify the above ONE statement to print two messages, such as ‘Message from ME’, and ‘Hello IIIT Delhi’ **on two different lines, but with an extra blank line in between.** HINT: ask yourself what would be the output if you were to simply execute the statement **print(‘\n’).**

2. Evaluating simple expressions: Write simple programs to evaluate arithmetic, relational or logical expressions. Here are some expressions. But, before you write the programs or statements be sure about what is it that you expect to be the result.
    - 2^3^4
    - 2^(3^4)
    - (2^3)^4
    - 2/3/4
    - 6 / (2 / 3)
    - 36 mod 15
    - 5 > 3 and 5.0 < 3.5
    - (not True) and False

3. Get some experience in writing input statements. Here are some simple exercises:
    - 3 time instants, T1, T2, T3 using multiple input statements, and **convert** the strings you have input into floating point numbers.
    - 3 time instants, T1, T2, T3 using ONE input statement, **BUT** without the use of split function.
    - 3 time instants, T1, T2, T3 using ONE input statement **and** using the split function
    -

4. Get some experience with objects with different data types, and with converting objects of one type to another type.
    a. Print the data type of:
    ‘IIITD’
    56.05
    ’29-08-2023’
    7 > 5
    36 mod 15
    5 > 3 and 5.0 < 3.5
    (not True) and False
    b. Convert
    ‘56.05’ to **int** data type
    56 to **float** data type
    ’29-08-2023’ to **int** data type
    True to **int** data type
    (5 < 3) to **int** data type

5. Write Python statements to print special characters:
    - ğ
    - क
    - ॐ
    - १
    - Symbols that represent the currencies of Europe, Japan, India (e.g. $ is US currency)

6. Write Python statements that will print the following strings, or a collection of strings that look like a figure:
    - ********** (of length 10)
    - --**--**--**--

    -
      ```
      ******
      *****
      ****
      ***
      **
      *
      ```
    -

7. Write Python statements to print some of the characters in the given string S = ‘Jai Jawan, Jai Kisan’:
    - Only the first part, viz. ‘Jai Jawan’
    - Only the second part, viz ‘Jai Kisan’
    - Just the words ‘Jawan’ and ‘Kisan’

8. Here are some algorithms involving conditional statements. Rewrite these as Python programs **and test them out**.
    - Below T1, T2, etc. are floating point nos.:
      ```
      input(T1, T2, T3, T4);
      minT = T1;
      if (T2 < minT) then minT = T2;
      if (T3 < minT) then minT = T3;
      if (T4 < minT) then minT = T4;
      output(minT)
      ```

    - Below T1 and T2 are integers :
      ```
      Input(T1, T2);
      if(T1 < T2) then minT = T1 else minT = T2;
      Print(minT)
      ```

    - Below INC and Tax are integers:
      ```
      input(INC); Tax = 0
      if INC > 200000 then [Tax = 10000 + 0.2*(INC-200000)]
      else [if INC > 100000 then [Tax = 0.1*(INC-100000)]]
      print(INC, Tax)
      ```

9. Write programs and test for the following problems (some have been discussed ib class):
    - Find the largest n such that 2^n ≤ 50.
    - Finding the SQRT(x) using two different methods.
    - Find the smallest integer < 100 that is divisible by 7 and 5.
    - Determine whether a given integer N is a prime number or not.