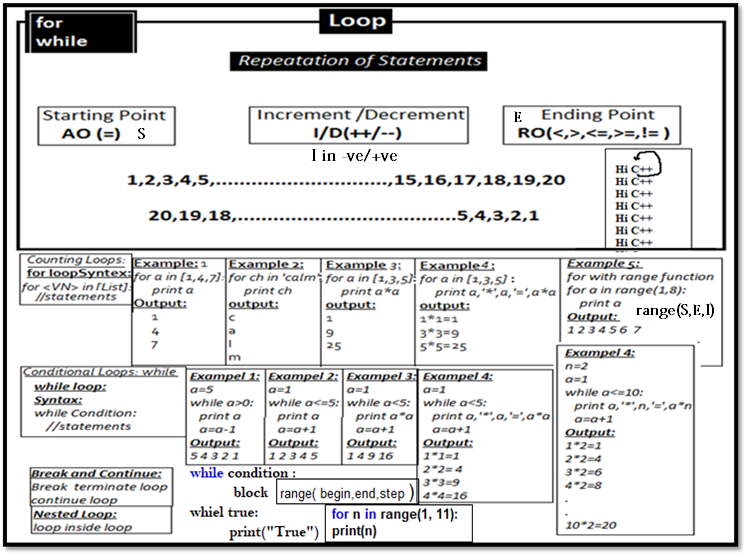
***Looping / Iteration*** (Repetition of statements)



***Iteration (Loop – For ,While, Do While, Nested)***

**Set 1**

1. WAP to display all number from 1 to 15.
2. WAP to display sum of all numbers from 1 to 10.
3. WAP to display Average of numbers from 1 to 6.
4. WAP to display multiply all numbers from 1 to 10.
5. WAP to display Square all numbers from 1to 7.
6. WAP to display Cube all numbers from 1to 8.
7. WAP to display all odd numbers from 1to 8
8. WAP to display all even numbers from 1to 8

**Set 2**

1. WAP to display all numbers from 1 to N.
2. WAP to display Sum number from 1 to N.
3. WAP to display Average number from 1 to N.
4. WAP to display Multiply number from 1 to N.
5. WAP to display Square Number from 1 to N.
6. WAP to display Cube all numbers from 1to N.
7. WAP to display all odd numbers from 1to N
8. WAP to display all even numbers from 1to N
9. WAP to display all 2 digit numbers from 1to N

**Set 3**

1. WAP to display of numbers from N to M.
2. WAP to display Sum of numbers from N to M.
3. WAP to display Average of numbers from N to M.
4. WAP to display multiply of numbers from N to M.
5. WAP to display Square of numbers from N to M.
6. WAP to display Cube number from N to M.
7. WAP to display all odd numbers from N to M.
8. WAP to display all even numbers from N to M.
9. WAP to display all 2 digit numbers from N to M.

**Set 4**

1. WAP to display even factors of a given number till N.
2. WAP to display odd factors of a given number till N.
3. WAP to display factorial of a given numbers.
4. WAP to check given number is prime or not
5. WAP to display all prime numbers from N to M.
6. WAP to check given two number are twin prime or not
7. WAP to check given numbers is perfect no or not.

**Set 5**

1. WAP to display result **xn.**
2. WAP to display result **(x+1)n**
3. WAP to display A if **A=P[1+r/100]n**
4. WAP to display Even and Odd numbers from N to M.
5. WAP to count Even and Odd numbers from N to M.
6. WAP to Add Even and Odd numbers in different variable from N to M.
7. WAP to calculate HCF(GCD) and LCM of two numbers
8. WAP to check max, min number in given 10 numbers

**Set 6**

1. WAP to display Reverse of a given numbers.
2. WAP to sum of all digit of a given numbers.
3. WAP to sum of all Even digit of a given numbers
4. WAP to sum of all odd digit of a given numbers
5. WAP to sum of all prime digit of a given numbers
6. WAP to print of all digit of a given numbers in different line
7. WAP to check given number is palindrome or not.121
8. WAP to check given number is Armstrom or not.153
9. WAP to check given number is duck or not.1034
10. WAP to check given number is Nelson or not.111,555
11. WAP to check given number is unique number or not.
12. WAP to check given number is BUZZ or not. Endwith and divisible by 7
13. WAP to check given number is Composite number or not.(A number said to be a composite number if its has one or more then one factor excluding 1 ant the number itself like 4,6,8,9,…..)
14. WAP to Find smallest digit of a number.
15. WAP to Find highest digit of a number.
16. WAP to check a numbers is strong numbers or not
17. WAP to check all digit of numbers ascending order or not.
18. WAP to check all digit of numbers descending order or not.

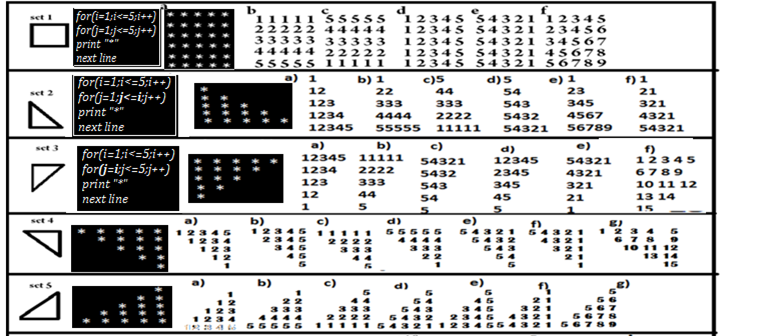
**Set 7**

1. WAP to display prime number between 1 to 1000
2. WAP to display perfect number between 1 to 100
3. WAP to find out all the Armstrom numbers between 1 to 1000
4. WAP to find out all the perfect numbers between 1 to 1000
5. WAP to find out all the palindrome number between 100 and 500.

**Set 8**

**6. WAP to print the following series**

1. 2,4,6,8,10……………n
2. 1,5,9,13……………….n
3. 1,4,9,16……………….n
4. 1,2,4,7,11,………………n
5. 1,8,27,64……………..n
6. 0,3,8,15,24,…..n
7. 3,6,9,12,…………….n
8. 2,5,10,17…………n
9. 0,7,26,……………n
10. 1,9,25,49,………………n
11. 0,3,8,15,……………..n
12. 24,99,224,399,………n



**sum = 0 # Initialize sum**

**for i in range(1, 100):**

**sum += i**

**print(sum)**

The following examples show how range can be used to produce a variety of sequences:

1. range(10) ->0,1,2,3,4,5,6,7,8,9
2. range(1, 10) ->1,2,3,4,5,6,7,8,9
3. range(1, 10, 2) ->1,3,5,7,9
4. range(10, 0, -1) ->10,9,8,7,6,5,4,3,2,1
5. range(10, 0, -2) ->10,8,6,4,2
6. range(2, 11, 2) ->2,4,6,8,10
7. range(-5, 5) -> −5,−4,−3,−2,−1,0,1,2,3,4
8. range(1, 2) ->1
9. range(1, 1) ->(empty)
10. range(1, -1) ->(empty)
11. range(1, -1, -1) ->1,0
12. range(0) ->(empty)

4. How many asterisks does the following code fragment print?

**a = 0**

**while a < 100:**

**print('\*', end='')**

**a += 1**

**print()**

5. How many asterisks does the following code fragment print?

**a = 0**

**while a < 100:**

**print('\*', end='')**

**print()**

6. How many asterisks does the following code fragment print?

**a = 0**

**while a > 100:**

**print('\*', end='')**

**a += 1**

**print()**

7. How many asterisks does the following code fragment print?

**a = 0**

**while a < 100:**

**b = 0**

**while b < 55:**

**print('\*', end='')**

**b += 1**

**print()**

**a += 1**

8. How many asterisks does the following code fragment print?

**a = 0**

**while a < 100:**

**if a % 5 == 0:**

**print('\*', end='')**

**a += 1**

**print()**

9. How many asterisks does the following code fragment print?

**a = 0**

**while a < 100:**

**b = 0**

**while b < 40:**

**if (a + b) % 2 == 0:**

**print('\*', end='')**

**b += 1**

**print()**

**a += 1**

10. How many asterisks does the following code fragment print?

**a = 0**

**while a < 100:**

**b = 0**

**while b < 100:**

**c = 0**

**while c < 100:**

**print('\*', end='')**

**c++**

**b += 1**

**a += 1**

**print()**

11. How many asterisks does the following code fragment print?

**for a in range(100):**

**print('\*', end='')**

**print()**

12. How many asterisks does the following code fragment print?

**for a in range(20, 100, 5):**

**print('\*', end='')**

**print()**

13. How many asterisks does the following code fragment print?

**for a in range(100, 0, -2):**

**print('\*', end='')**

**print()**

14. How many asterisks does the following code fragment print?

**for a in range(1, 1):**

**print('\*', end='')**

**print()**

15. How many asterisks does the following code fragment print?

**for a in range(-100, 100):**

**print('\*', end='')**

**print()**

16. How many asterisks does the following code fragment print?

**for a in range(-100, 100, 10):**

**print('\*', end='')**

**print()**

17. Rewrite the code in the previous question so it uses a while instead of a for. Your code should

behave identically.

18. How many asterisks does the following code fragment print?

**for a in range(-100, 100, -10):**

**print('\*', end='')**

**print()**

19. How many asterisks does the following code fragment print?

**for a in range(100, -100, 10):**

**print('\*', end='')**

**print()**

20. How many asterisks does the following code fragment print?

**for a in range(100, -100, -10):**

**print('\*', end='')**

**print()**

21. What is printed by the following code fragment?

**a = 0**

**while a < 100:**

**print(a)**

**a += 1**

**print()**

22. Rewrite the code in the previous question so it uses a for instead of a while. Your code should

behave identically.

23. What is printed by the following code fragment?

**a = 0**

**while a > 100:**

**print(a)**

**a += 1**

**print()**

24. Rewrite the following code fragment using a break statement and eliminating the done variable.

Your code should behave identically to this code fragment.

**done = False**

**n, m = 0, 100**

**while not done and n != m:**

**n = eval(input())**

**if n < 0:**

**done = true**

**print("n =", n)**

25. Rewrite the following code fragment so it does not use a break statement. Your code should behave

identically to this code fragment.

**// Code with break ...**

26. Rewrite the following code fragment so it eliminates the continue statement. Your new code’s logic

should be simpler than the logic of this fragment.

**x = 100**

**while x > 0:**

**y = eval(input())**

**if y == 25:**

**x += 1**

**continue**

**x = eval(input())**

**print('x =', x)**

27. What is printed by the following code fragment?

**a = 0**

**while a < 100:**

**print(a, end='')**

**a += 1**

**print()**