1.If conditional statement

In [107…
```
Condition = True
if Condition:
    j_anger = True
    s_anger = True
else:
    j_anger = False
    s_anger = False

print("j_anger:", j_anger)
print("s_anger:", s_anger)
```

```
j_anger: True
s_anger: True
```

In [108…
```
Condition = True
if Condition:
    j_anger = True
    s_anger = False
else:
    j_anger = True
    s_anger = False

print("j_anger:", j_anger)
print("s_anger:", s_anger)
```

```
j_anger: True
s_anger: False
```

2.Mark Even and Odd

In [109…
```
x = 4

r = x % 2

if r == 0:
    print('Even number')

if r == 1:
    print('odd number')
```

```
Even number
```

In [110…
```
x = 5

r = x % 2

if r == 0:
    print('Even number')

if r == 1:
    print('odd number')
```

```
odd number
```

In [111…
```
def checkOddEven(number):
    if number % 2 == 0:
        return True
```

```
    else:
        return False
print(checkOddEven(4))
print(checkOddEven(5))
```

```
True
False
```

3.The Else Statement

In [112…
```
def friends_in_trouble(a_smile, b_smile):
    if a_smile == b_smile:
        return True
    else:
        return False
print(friends_in_trouble(True, True))
print(friends_in_trouble(False, False))
print(friends_in_trouble(True, False))
```

```
True
True
False
```

4.Cat and Hat

In [113…
```
def cat_hat(str):
    return str.replace("cat", "").replace("hat", "")
```

In [114…
```
print(cat_hat("cat in the hat"))       # Output: " in the "
print(cat_hat("catch a hat and cat"))  # Output: "ch a  and "
print(cat_hat("nothing to remove"))
```

```
 in the
ch a  and
nothing to remove
```

5.The FizzBuzz Program

In [115…
```
def fizz_buzz(a):
    if a % 3 == 0 and a % 5 == 0:
        print("FizzBuzz")
    elif a % 3 == 0:
        print("Fizz")
    elif a % 5 == 0:
        print("Buzz")
    else:
        print(a)
```

In [116…
```
fizz_buzz(3)
fizz_buzz(5)
fizz_buzz(15)
fizz_buzz(7)
```

```
Fizz
Buzz
FizzBuzz
7
```

6.Check the status

In [117…
```
def check_status(a, b, flag):
    if ((a >= 0) ^ (b >= 0)) and flag == False:
        return True
    elif a < 0 and b < 0 and flag == True:
        return True
    else:
        return False
```

In [118…
```
print(check_status(1, -1, False))
print(check_status(-182, -9121, True))
print(check_status(5, 3, True))
print(check_status(-5, 6, False))
print(check_status(6, 6, False))
```

```
True
True
False
True
False
```

7.Even Odd Game

In [119…
```
def who_wins(n):
    if n % 2 == 1:
        print("You")
    else:
        print("Friend")
```

In [120…
```
who_wins(9)
who_wins(4)
who_wins(1)
who_wins(2)
```

```
You
Friend
You
Friend
```

8.Odd or Even

In [121…
```
def is_even(n):
    return n % 2 == 0
```

In [122…
```
print(is_even(15))
print(is_even(44))
```

```
False
True
```

In [123…
```
def is_odd(n):
    return n % 2 != 0
```

In [124…
```
print(is_even(11))
print(is_even(40))
```

```
False
True
```

9.Greatest of Three

In [125…
```
def find_greatest(a, b, c):
    return max(a, b, c)
```

In [126…
```
print(find_greatest(1, 2, 3))
print(find_greatest(2, 2, 5))
print(find_greatest(100, 300, 200))
```

```
3
5
300
```

10.Leap Year

In [127…
```
def is_leap_year(year):
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        print("True")
    else:
        print("False")
```

In [128…
```
is_leap_year(1900)
is_leap_year(2020)
```

```
False
True
```

11.Calculator

In [129…
```
def basic_operations(a, b, operator):
    if operator == 1:
        print(a + b, end="")
    elif operator == 2:
        print(a - b, end="")
    elif operator == 3:
        print(a * b, end="")
    else:
        print("Invalid Input", end="")
```

In [130…
```
basic_operations(1,2,3)
```

```
2
```

In [131…
```
basic_operations(2,2,2)
```

```
0
```

In [132…
```
basic_operations(5,4,7)
```

```
Invalid Input
```

12.Closest Number

In [133…
```
def closest_divisible(n, m):
    q = n // m
    n1 = m * q
    n2 = m * (q + 1)
    n3 = m * (q - 1)
    candidates = [n1]
```

```
    if n != n1:
        candidates.append(n2)
        candidates.append(n3)
    return min(candidates, key=lambda x: (abs(n - x), -abs(x)))
```

In [134…
```
print(closest_divisible(13, 4))
print(closest_divisible(-15, 6))
print(closest_divisible(7, 3))
print(closest_divisible(-7, 3))
```

```
12
-18
6
-6
```

13.Dice Problem

In [135…
```
def opposite_face(n):
    if 1 <= n <= 6:
        return 7 - n
    else:
        return "Invalid input. Enter a number from 1 to 6."
```

In [136…
```
print(opposite_face(6))
print(opposite_face(2))
print(opposite_face(4))
```

```
1
5
3
```

14.Valid Triangle

In [137…
```
def do_overlap(L1, R1, L2, R2):
    if L1[0] > R2[0] or L2[0] > R1[0]:
        return 0
    if R1[1] > L2[1] or R2[1] > L1[1]:
        return 0

    return 1
```

In [138…
```
# Test Case 1
print(do_overlap((0,10), (10,0), (5,5), (15,0)))
# Test Case 2
print(do_overlap((0,2), (1,1), (-2,0), (0,-3)))
```

```
1
0
```

15.Test if tuple is distinct

In [139…
```
def all_unique(arr):
    print(len(arr) == len(set(arr)))
```

In [140…
```
all_unique((1, 2, 3, 4, 5, 4))
all_unique((1, 2, 3, 4, 5))
```

```
False
True
```

16.Day before N days

In [141…
```
def day_n_days_before(d, n):
    return (d - n) % 7
if __name__ == "__main__":
    d = int(input("Enter the day index (0=Sunday, ..., 6=Saturday): "))
    n = int(input("Enter the number of days before: "))
    result = day_n_days_before(d, n)
    print("Output:", result)
```

```
Output: 1
```

17.Solving queries

In [142…
```
def query_dictionary(d, queries):
    for key in queries:
        print(d.get(key, "None"))
d = {1: "abc", 2: "cde", 3: "fgh"}
queries = [2, 3, 4]
query_dictionary(d, queries)
```

```
cde
fgh
None
```

18.Factorial

In [143…
```
def factorial(n):
    if n == 0:
        return 1
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result

# Example usage
n = int(input("Enter a number (0 to 10): "))
if 0 <= n <= 10:
    print("Factorial:", factorial(n))
else:
    print("Invalid input. Please enter a number between 0 and 10.")
```

```
Factorial: 24
```

19.Check Prime

In [144…
```
def is_prime(n):
    if n <= 1:
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
n = int(input("Enter a number: "))
print(is_prime(n))
```

```
True
```

In [145…
```
def is_prime(n):
    if n <= 1:
```

```
        return False
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            return False
    return True
n = int(input("Enter a number: "))
print(is_prime(n))
```

```
False
```

20.Next Prime Number

In [146…

```
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

def next_prime(n):
    candidate = n + 1
    while True:
        if is_prime(candidate):
            return candidate
        candidate += 1
n = int(input("Enter a number: "))
print(next_prime(n))
```

```
17
```

21,Fibonacci Number

In [147…

```
def fibonacci(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1

    a, b = 0, 1
    for _ in range(2, n + 1):
        a, b = b, a + b
    return b

# Example usage
n = int(input("Enter the value of n: "))
print(fibonacci(n))
```

```
3
```

22.Perfect Number

In [148…

```
def is_perfect(n):
    if n == 1:
        return False
    sum_factors = 1
    i = 2
    while i * i <= n:
        if n % i == 0:
            sum_factors += i
```

```
            if i != n // i:
                sum_factors += n // i
        i += 1

    return sum_factors == n
n = int(input("Enter a number: "))
print(is_perfect(n))
```

```
False
```

In [149…

```
def is_perfect(n):
    if n == 1:
        return False
    sum_factors = 1
    i = 2
    while i * i <= n:
        if n % i == 0:
            sum_factors += i
            if i != n // i:
                sum_factors += n // i
        i += 1

    return sum_factors == n
n = int(input("Enter a number: "))
print(is_perfect(n))
```

```
True
```

23.Adam Number

In [150…

```
def checkAdamOrNot(N):
    def reverse_num(num):
        return int(str(num)[::-1])

    rev_N = reverse_num(N)
    N_sq = N * N
    rev_N_sq = rev_N * rev_N

    if reverse_num(N_sq) == rev_N_sq:
        return "YES"
    else:
        return "NO"
```

In [151…

```
print(checkAdamOrNot(12))
print(checkAdamOrNot(14))
```

```
YES
NO
```

24.Check Strong Number

In [152…

```
def isStrong(N):
    def factorial(n):
        fact = 1
        for i in range(2, n+1):
            fact *= i
        return fact

    original = N
    total = 0
```

```
    while N > 0:
        digit = N % 10
        total += factorial(digit)
        N //= 10

    return 1 if total == original else 0
```

In [153…

```
print(isStrong(145))
print(isStrong(14))
```

```
1
0
```

In [ ]: