**Question 1**

|  |
| --- |
| **Source Code** |

|  |
| --- |
| # Write a program to demonstrate the use of different operators in python.  def operators\_demo(a, b):  print("Addition:", a + b)  print("Subtraction:", a - b)  print("Multiplication:", a \* b)  print("Division:", a / b if b != 0 else "Undefined")  print("Modulus:", a % b if b != 0 else "Undefined")  print("Floor Division:", a // b if b != 0 else "Undefined")  print("Exponentiation:", a \*\* b)  print("Equal:", a == b)  print("Not Equal:", a != b)  print("Greater:", a > b)  print("Smaller:", a < b)  print("Logical AND:", a > 0 and b > 0)  print("Logical OR:", a > 0 or b > 0)  print("Logical NOT:", not(a > 0))  a = int(input("Enter first number: ")) b = int(input("Enter second number: ")) operators\_demo(a, b) |

|  |
| --- |
| **Output** |

|  |
| --- |
| Enter first number: Traceback (most recent call last):  File "<string>", line 17, in <module> EOFError: EOF when reading a line |

**Question 2**

|  |
| --- |
| **Source Code** |

|  |
| --- |
| # Write a program to print Fibonacci Series 0 1 1 2 3 5 ………..N  n = int(input("Enter the number of terms: ")) a, b = 0, 1 print("Fibonacci Series:") for \_ in range(n):  print(a)  a, b = b, a + b |

|  |
| --- |
| **Output** |

|  |
| --- |
| Enter the number of terms:  Traceback (most recent call last):  File "<string>", line 4, in <module> EOFError: EOF when reading a line |

**Question 3**

|  |
| --- |
| **Source Code** |

|  |
| --- |
| # Write a program to print the sum of first n prime numbers.  def is\_prime(num):  if num < 2:  return False  for i in range(2, int(num \*\* 0.5) + 1):  if num % i == 0:  return False  return True  n = int(input("Enter Number of Prime Numbers to Sum: ")) count, num, total = 0, 2, 0 while count < n:  if is\_prime(num):  total += num  count += 1  num += 1 print("Sum of first", n, "prime numbers is:", total) |

|  |
| --- |
| **Output** |

|  |
| --- |
| Enter Number of Prime Numbers to Sum: Traceback (most recent call last):  File "<string>", line 10, in <module> EOFError: EOF when reading a line |