Name: Aditi M.Tech. CSA Year 1

Timestamp: 08/01/2025 11:11

Q1: Write a program that prints all the numbers from 1 to 100. For multiples of 3, print "Fizz" instead of the number, and for multiples of 5, print "Buzz". For numbers that are multiples of both 3 and 5, print "FizzBuzz". Any othernumber print as it is.

In [1]:
```
for i in range(1,101):
    if i%3==0 and i%5==0:
        print("Fizzbuzz", end =" ")
    elif i%3==0 and i%5!=0:
        print("Fizz", end =" ")
    elif i%5==0 and i%3!=0:
        print("Buzz", end =" ")
    else:
        print(i, end =" ")
```

```
1 2 Fizz 4 Buzz Fizz 7 8 Fizz Buzz 11 Fizz 13 14 Fizzbuzz 16 17 Fizz 19 Buzz
Fizz 22 23 Fizz Buzz 26 Fizz 28 29 Fizzbuzz 31 32 Fizz 34 Buzz Fizz 37 38 Fiz
z Buzz 41 Fizz 43 44 Fizzbuzz 46 47 Fizz 49 Buzz Fizz 52 53 Fizz Buzz 56 Fizz
58 59 Fizzbuzz 61 62 Fizz 64 Buzz Fizz 67 68 Fizz Buzz 71 Fizz 73 74 Fizzbuzz
76 77 Fizz 79 Buzz Fizz 82 83 Fizz Buzz 86 Fizz 88 89 Fizzbuzz 91 92 Fizz 94
Buzz Fizz 97 98 Fizz Buzz
```

Q2: Write a recursive function factorial(n) to calculate the factorial of a given positive integer n.

In [2]:
```
def factorial(n):
    if n==0:
        return 1
    return n*factorial(n-1)
```

In [3]:
```
#eg
num = 5
print(factorial(num))
```

```
120
```

Q3: Define a class BankAccount in python program with the following: Attributes: account_number, account_holder, balance. Methods: deposit(amount): Adds the amount to the balance. withdraw(amount): Subtracts the amount from the balance if sufficient funds are available. get_balance(): Returns the current balance. Demonstrate the use of this class with an example.

In [4]:
```
class BankAccount:
    def __init__(self):
        self.bal = 0
    def get_balance(self):
        print("Current balance is:", self.bal)
    def deposit(self,amt):
        self.bal = self.bal + amt
    def withdraw(self,amt):
        if amt>self.bal:
            print("Insufficient")
        else:
            self.bal = self.bal - amt
```

In [5]:
```
b = BankAccount()
```

In [6]:
```
b.deposit(1000)
b.get_balance()
```

```
Current balance is: 1000
```

In [7]:
```
b.withdraw(500)
b.get_balance()
```

```
Current balance is: 500
```

In [8]:
```
b.withdraw(600)
b.get_balance()
```

```
Insufficient
Current balance is: 500
```

In [9]:
```
b.get_balance()
```

```
Current balance is: 500
```

Q4: Using the math library to write a function calculate_distance(x1, y1, x2, y2) that calculates the Euclidean distance between two points (x1, y1) and (x2, y2).

In [10]:
```
import math
def calculate_distance(x1,y1,x2,y2):
    return math.sqrt(math.pow(x1-x2,2) + math.pow(y1-y2,2))
```

In [11]:
```
#eg 0,0 and 3,4
calculate_distance(0,0,3,4)
```

Out[11]:
```
5.0
```

Q5: Write a Python program that uses list comprehension to generate a list of all numbers between 1 and 1000 divisible by 7 but not by 5.

In [12]:
```
res = [x for x in range(1,1001) if x%5!=0 and x%7==0]
print(res)
```

```
[7, 14, 21, 28, 42, 49, 56, 63, 77, 84, 91, 98, 112, 119, 126, 133, 147, 154,
161, 168, 182, 189, 196, 203, 217, 224, 231, 238, 252, 259, 266, 273, 287, 29
4, 301, 308, 322, 329, 336, 343, 357, 364, 371, 378, 392, 399, 406, 413, 427,
434, 441, 448, 462, 469, 476, 483, 497, 504, 511, 518, 532, 539, 546, 553, 56
7, 574, 581, 588, 602, 609, 616, 623, 637, 644, 651, 658, 672, 679, 686, 693,
707, 714, 721, 728, 742, 749, 756, 763, 777, 784, 791, 798, 812, 819, 826, 83
3, 847, 854, 861, 868, 882, 889, 896, 903, 917, 924, 931, 938, 952, 959, 966,
973, 987, 994]
```

Q6: Write a Python function remove_duplicates(lst) that takes a list as input and returns a new list with all duplicate elements removed, maintaining the original order without inbuilt functions.

In [13]:
```
def remove_duplicates(lst):
    new=[]
    for i in range(len(lst)):
        if lst[i] not in new:
            new.append(lst[i])
    return new
```

In [14]:
```
l = [1,1,2,3,3,3,4]
remove_duplicates(l)
```

Out[14]:
```
[1, 2, 3, 4]
```