**Innovatily**

**2.Pandas: DataFrame Manipulation**

**Question:**

**Given a Pandas DataFrame containing sales data, perform the following operations:**

**import pandas as pd**

**data = {**

**'Product': ['A', 'B', 'A', 'C', 'B', 'C'],**

**'Sales': [100, 200, 150, 300, 250, 400],**

**'Region': ['North', 'South', 'North', 'East', 'South', 'East']}**

1. Find the total sales per product.

* **df.groupby('Product')['Sales'].sum()**

1. Find the highest sales value per region.

* **df.groupby('Region')['Sales'].max()**

1. Add a new column Discounted Price which is 90% of the sales value.

* **df['Discounted Price'] = df['Sales']\*0.9**
* **df**

**3. Data Structures: Reverse Words in a String**

**Question:**

**Write a function that reverses the words in a given string. Do not use built-in reverse() or slicing.**

**Example:**

**Input: "Python is great"**

**Output: "great is Python"**

Answer:

def reverse\_words(s):

    words = s.split()

    reverse\_words = words[::-1]

    return " ".join(reverse\_words)

s = "Python is great"

print(reverse\_words(s))

**4. Algorithm: Find Missing Number in an Arrary**

**Question:**

**Given a list of n-1 numbers in the range 1 to n, find the missing number.**

**Example:**

**Input: [1, 2, 4, 5, 6]**

**Output: 3**

**Answer:**

def missing\_number(s):

    n = len(s)+1

    total\_sum = n\*(n+1)//2

    actual\_sum = sum(s)

    return total\_sum - actual\_sum

s = [1, 2, 4, 5, 6]

print(missing\_number(s))