**Task-4:** **Applying Custom NumPy Universal Functions**

1. **Description**

Create a custom numpy ufunc (universal function) and apply it to a numpy array.

1. **Output**

****

1. **Algorithm Used in Task**

This code demonstrates how to create and use a custom universal function (ufunc) in NumPy. By converting a Python function to a ufunc using np.frompyfunc, the function can be applied efficiently to each element of a NumPy array. This approach is useful for applying complex, custom operations element-wise on large datasets.

**Explanation of Libraries and Their Usage:**

1. **NumPy (import numpy as np)**
   * A library for numerical computing in Python, enabling fast operations on arrays and matrices.
   * **Key Function Used:**
     + np.frompyfunc: Converts a standard Python function into a universal function (ufunc) that works element-wise on NumPy arrays.

**Approach:**

1. Define a custom Python function (custom\_func) to perform mathematical operations.
2. Use NumPy's frompyfunc to convert the custom function into a ufunc.
3. Apply the ufunc element-wise to a NumPy array.
4. Display the original array and the transformed result.

**Algorithm:**

1. Import the NumPy library.
2. Define a Python function (custom\_func) to compute x2+5 for an input x.
3. Use np.frompyfunc(custom\_func, 1, 1) to convert the Python function into a ufunc:
   * The first argument is the Python function to convert.
   * The second argument specifies the number of input arguments the function takes (1 in this case).
   * The third argument specifies the number of outputs the function returns (1 here).
4. Create a NumPy array containing numerical values.
5. Apply the ufunc to the array using custom\_ufunc(arr).
6. Print the original array and the transformed result.