In [62]:
```
"""1)   Define the following metrics and perform the following operations
i)      Write a Python program using Python Lists
ii)     Write a Python program and NumPy

Matrix A =  [[ 3.7827  3.3454  3.2341]  , [ 2.2122  3.5678  3.9087] ,
[1.1234  2.8934,  5.9087]].

Matrix B =  [[ 3.1234  3.0987  3.1234]  , [ 2.1111  3.2222  3.3333] ,
[1.0987  1.3456,  5.1234]].

Matrix C =  [[ 3.1243  3.0989  3.1256 ]  , [ 2.6721  3.6785  3.9017] ,
[1.1254  2.8956,  5.9187]]."""

Matrix_A=  [[3.7827, 3.3454,  3.2341]  , [ 2.2122, 3.5678 , 3.9087] ,
[1.1234 , 2.8934,  5.9087]]

Matrix_B =  [[3.1234 , 3.0987 , 3.1234]  , [2.1111 , 3.2222 , 3.3333] ,
[1.0987 , 1.3456,  5.1234]]

Matrix_C =  [[ 3.1243 ,3.0989 , 3.1256 ]  , [2.6721 , 3.6785 , 3.9017] ,
[1.1254 , 2.8956,  5.9187]]
print("Matrix_A")
print(Matrix_A)
print("Matrix_B")
print(Matrix_B)
print("Matrix_C")
print(Matrix_C)
```

```
Matrix_A
[[3.7827, 3.3454, 3.2341], [2.2122, 3.5678, 3.9087], [1.1234, 2.8934, 5.9087]]
Matrix_B
[[3.1234, 3.0987, 3.1234], [2.1111, 3.2222, 3.3333], [1.0987, 1.3456, 5.1234]]
Matrix_C
[[3.1243, 3.0989, 3.1256], [2.6721, 3.6785, 3.9017], [1.1254, 2.8956, 5.9187]]
```

In [64]:
```
"""ii)  Write a Python program and NumPy"""
import numpy as np
def matrixs(values):
    return np.array(values)

A_1 = matrixs([[3.7827, 3.3454, 3.2341],
                   [2.2122, 3.5678, 3.9087],
                   [1.1234, 2.8934, 5.9087]])

B_1 = matrixs([[3.1234, 3.0987, 3.1234],
                   [2.1111, 3.2222, 3.3333],
                   [1.0987, 1.3456, 5.1234]])

C_1 = matrixs([[3.1243, 3.0989, 3.1256],
                   [2.6721, 3.6785, 3.9017],
                   [1.1254, 2.8956, 5.9187]])


matrices = {'A': A_1, 'B': B_1, 'C': C_1}
```

```
for name, matrix in matrices.items():
    print(f"Matrix {name}")
    print(matrix)
    print()
```

```
Matrix A
[[3.7827 3.3454 3.2341]
 [2.2122 3.5678 3.9087]
 [1.1234 2.8934 5.9087]]

Matrix B
[[3.1234 3.0987 3.1234]
 [2.1111 3.2222 3.3333]
 [1.0987 1.3456 5.1234]]

Matrix C
[[3.1243 3.0989 3.1256]
 [2.6721 3.6785 3.9017]
 [1.1254 2.8956 5.9187]]
```

In [66]:
```
""" 2.  Write a Python Program and perform the following operations ?
I.      Create a List {1225, 4986, 6789, 7890, 2345, 6783, 0987, 1234, 8765, 3456}"
numlist = [1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]

print("Original List:")
print(numlist)
```

```
Original List:
[1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]
```

In [68]:
```
""" II. Iterate using a for loop"""

numlist = [1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]
print("Original Number List:")
print(numlist)

print("\nIterating through the list and printing each number:")
for number in numlist:
    print(number)
```

```
Original Number List:
[1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]

Iterating through the list and printing each number:
1225
4986
6789
7890
2345
6783
987
1234
8765
3456
```

In [70]:
```
""" III.        Iterate using for loop and range """

numlist = [1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]
print("Original Number List:")
print(numlist)

print("\nIterating through loop:")
for i in range(len(numlist)):
    print(f"Index {i}: {numlist[i]}")
```

```
Original Number List:
[1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]

Iterating through loop:
Index 0: 1225
Index 1: 4986
Index 2: 6789
Index 3: 7890
Index 4: 2345
Index 5: 6783
Index 6: 987
Index 7: 1234
Index 8: 8765
Index 9: 3456
```

In [72]:
```
""" IV. List Comprehension"""
numlist = [1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]
list1 = [number for number in numlist]

print("New List:")
print(list1)
```

```
New List:
[1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]
```

In [74]:
```
""" V.  Enumerate"""
numlist = [1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]
print("Enumerating through the list:")
for index, number in enumerate(numlist):
    print(f"Index {index}: {number}")
```

```
Enumerating through the list:
Index 0: 1225
Index 1: 4986
Index 2: 6789
Index 3: 7890
Index 4: 2345
Index 5: 6783
Index 6: 987
Index 7: 1234
Index 8: 8765
Index 9: 3456
```

In [76]:
```
""" VI. Iter function and next function"""
numlist = [1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]
numberiterator = iter(numlist)
```

```
print("Iterating through the list using iter() and next():")


print(next(numberiterator))

for number in numberiterator:
    print(number)
```

```
Iterating through the list using iter() and next():
1225
4986
6789
7890
2345
6783
987
1234
8765
3456
```

In [78]:
```
""" VII.        Map function"""
numlist = [1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]
def double_number(number):
    return number * 2

doubled_list = map(double_number, numlist)

doubled_list = list(doubled_list)
print("List of doubled numbers using map():")
print(doubled_list)
```

```
List of doubled numbers using map():
[2450, 9972, 13578, 15780, 4690, 13566, 1974, 2468, 17530, 6912]
```

In [80]:
```
""" VIII.       Using zip"""
numlist = [1225, 4986, 6789, 7890, 2345, 6783, 987, 1234, 8765, 3456]
NumList = [1,2,3,4,5,6,7,8,9,10]

combined_list = zip(numlist, NumList)

combined_list = list(combined_list)
print("Combined list using zip():")
print(combined_list)
```

```
Combined list using zip():
[(1225, 1), (4986, 2), (6789, 3), (7890, 4), (2345, 5), (6783, 6), (987, 7), (1234,
8), (8765, 9), (3456, 10)]
```

In [82]:
```
""" IX. Using NumPy Module """
numberArray = np.array(numlist)
print("NumPy array:")
print(numberArray)
```

```
NumPy array:
[1225 4986 6789 7890 2345 6783  987 1234 8765 3456]
```

In [84]:
```
""" 3)  For a List A, B, C, D, E write a python program to compute all the combinat
```

```python
def get_combinations(n):

    def helper_combinations(prefix, remaining):
        if not remaining:
            return [prefix]
        without_first = helper_combinations(prefix, remaining[1:])
        with_first = helper_combinations(prefix + [remaining[0]], remaining[1:])
        return with_first + without_first


    return helper_combinations([], n)[1:]

def get_permutations(n):

    def helper_permutations(n):
        if len(n) == 1:
            return [n]
        perms = []
        for i in range(len(n)):
            remaining_elements = n[:i] + n[i+1:]
            for p in helper_permutations(remaining_elements):
                 perms.append([n[i]] + p)
        return perms

    return helper_permutations(n)


input_list = ["A", "B", "C", "D", "E"]


print("Combinations\n")
combinations = get_combinations(input_list)
for combination in combinations:
    print(combination)

print("\n\n")
print("Permutations\n")
permutations = get_permutations(input_list)
for permutation in permutations:
    print(permutation)
```

Combinations

```
['A', 'B', 'C', 'D']
['A', 'B', 'C', 'E']
['A', 'B', 'C']
['A', 'B', 'D', 'E']
['A', 'B', 'D']
['A', 'B', 'E']
['A', 'B']
['A', 'C', 'D', 'E']
['A', 'C', 'D']
['A', 'C', 'E']
['A', 'C']
['A', 'D', 'E']
['A', 'D']
['A', 'E']
['A']
['B', 'C', 'D', 'E']
['B', 'C', 'D']
['B', 'C', 'E']
['B', 'C']
['B', 'D', 'E']
['B', 'D']
['B', 'E']
['B']
['C', 'D', 'E']
['C', 'D']
['C', 'E']
['C']
['D', 'E']
['D']
['E']
[]
```

Permutations

```
['A', 'B', 'C', 'D', 'E']
['A', 'B', 'C', 'E', 'D']
['A', 'B', 'D', 'C', 'E']
['A', 'B', 'D', 'E', 'C']
['A', 'B', 'E', 'C', 'D']
['A', 'B', 'E', 'D', 'C']
['A', 'C', 'B', 'D', 'E']
['A', 'C', 'B', 'E', 'D']
['A', 'C', 'D', 'B', 'E']
['A', 'C', 'D', 'E', 'B']
['A', 'C', 'E', 'B', 'D']
['A', 'C', 'E', 'D', 'B']
['A', 'D', 'B', 'C', 'E']
['A', 'D', 'B', 'E', 'C']
['A', 'D', 'C', 'B', 'E']
['A', 'D', 'C', 'E', 'B']
['A', 'D', 'E', 'B', 'C']
['A', 'D', 'E', 'C', 'B']
```

['A', 'E', 'B', 'C', 'D']
['A', 'E', 'B', 'D', 'C']
['A', 'E', 'C', 'B', 'D']
['A', 'E', 'C', 'D', 'B']
['A', 'E', 'D', 'B', 'C']
['A', 'E', 'D', 'C', 'B']
['B', 'A', 'C', 'D', 'E']
['B', 'A', 'C', 'E', 'D']
['B', 'A', 'D', 'C', 'E']
['B', 'A', 'D', 'E', 'C']
['B', 'A', 'E', 'C', 'D']
['B', 'A', 'E', 'D', 'C']
['B', 'C', 'A', 'D', 'E']
['B', 'C', 'A', 'E', 'D']
['B', 'C', 'D', 'A', 'E']
['B', 'C', 'D', 'E', 'A']
['B', 'C', 'E', 'A', 'D']
['B', 'C', 'E', 'D', 'A']
['B', 'D', 'A', 'C', 'E']
['B', 'D', 'A', 'E', 'C']
['B', 'D', 'C', 'A', 'E']
['B', 'D', 'C', 'E', 'A']
['B', 'D', 'E', 'A', 'C']
['B', 'D', 'E', 'C', 'A']
['B', 'E', 'A', 'C', 'D']
['B', 'E', 'A', 'D', 'C']
['B', 'E', 'C', 'A', 'D']
['B', 'E', 'C', 'D', 'A']
['B', 'E', 'D', 'A', 'C']
['B', 'E', 'D', 'C', 'A']
['C', 'A', 'B', 'D', 'E']
['C', 'A', 'B', 'E', 'D']
['C', 'A', 'D', 'B', 'E']
['C', 'A', 'D', 'E', 'B']
['C', 'A', 'E', 'B', 'D']
['C', 'A', 'E', 'D', 'B']
['C', 'B', 'A', 'D', 'E']
['C', 'B', 'A', 'E', 'D']
['C', 'B', 'D', 'A', 'E']
['C', 'B', 'D', 'E', 'A']
['C', 'B', 'E', 'A', 'D']
['C', 'B', 'E', 'D', 'A']
['C', 'D', 'A', 'B', 'E']
['C', 'D', 'A', 'E', 'B']
['C', 'D', 'B', 'A', 'E']
['C', 'D', 'B', 'E', 'A']
['C', 'D', 'E', 'A', 'B']
['C', 'D', 'E', 'B', 'A']
['C', 'E', 'A', 'B', 'D']
['C', 'E', 'A', 'D', 'B']
['C', 'E', 'B', 'A', 'D']
['C', 'E', 'B', 'D', 'A']
['C', 'E', 'D', 'A', 'B']
['C', 'E', 'D', 'B', 'A']
['D', 'A', 'B', 'C', 'E']
['D', 'A', 'B', 'E', 'C']

```
['D', 'A', 'C', 'B', 'E']
['D', 'A', 'C', 'E', 'B']
['D', 'A', 'E', 'B', 'C']
['D', 'A', 'E', 'C', 'B']
['D', 'B', 'A', 'C', 'E']
['D', 'B', 'A', 'E', 'C']
['D', 'B', 'C', 'A', 'E']
['D', 'B', 'C', 'E', 'A']
['D', 'B', 'E', 'A', 'C']
['D', 'B', 'E', 'C', 'A']
['D', 'C', 'A', 'B', 'E']
['D', 'C', 'A', 'E', 'B']
['D', 'C', 'B', 'A', 'E']
['D', 'C', 'B', 'E', 'A']
['D', 'C', 'E', 'A', 'B']
['D', 'C', 'E', 'B', 'A']
['D', 'E', 'A', 'B', 'C']
['D', 'E', 'A', 'C', 'B']
['D', 'E', 'B', 'A', 'C']
['D', 'E', 'B', 'C', 'A']
['D', 'E', 'C', 'A', 'B']
['D', 'E', 'C', 'B', 'A']
['E', 'A', 'B', 'C', 'D']
['E', 'A', 'B', 'D', 'C']
['E', 'A', 'C', 'B', 'D']
['E', 'A', 'C', 'D', 'B']
['E', 'A', 'D', 'B', 'C']
['E', 'A', 'D', 'C', 'B']
['E', 'B', 'A', 'C', 'D']
['E', 'B', 'A', 'D', 'C']
['E', 'B', 'C', 'A', 'D']
['E', 'B', 'C', 'D', 'A']
['E', 'B', 'D', 'A', 'C']
['E', 'B', 'D', 'C', 'A']
['E', 'C', 'A', 'B', 'D']
['E', 'C', 'A', 'D', 'B']
['E', 'C', 'B', 'A', 'D']
['E', 'C', 'B', 'D', 'A']
['E', 'C', 'D', 'A', 'B']
['E', 'C', 'D', 'B', 'A']
['E', 'D', 'A', 'B', 'C']
['E', 'D', 'A', 'C', 'B']
['E', 'D', 'B', 'A', 'C']
['E', 'D', 'B', 'C', 'A']
['E', 'D', 'C', 'A', 'B']
['E', 'D', 'C', 'B', 'A']
```

In [86]:

```
""" 4)  Using the same list use itertools compute permutations and combinations """

import itertools


input_list = ['A', 'B', 'C', 'D', 'E']


print("Permutations:")
```

```
for length in range(1, len(input_list) + 1):
    perms = itertools.permutations(input_list, length)
    for perm in perms:
        print(perm)


print("\nCombinations:")
for length in range(1, len(input_list) + 1):
    combs = itertools.combinations(input_list, length)
    for comb in combs:
        print(comb)
```

```
Permutations:
('A',)
('B',)
('C',)
('D',)
('E',)
('A', 'B')
('A', 'C')
('A', 'D')
('A', 'E')
('B', 'A')
('B', 'C')
('B', 'D')
('B', 'E')
('C', 'A')
('C', 'B')
('C', 'D')
('C', 'E')
('D', 'A')
('D', 'B')
('D', 'C')
('D', 'E')
('E', 'A')
('E', 'B')
('E', 'C')
('E', 'D')
('A', 'B', 'C')
('A', 'B', 'D')
('A', 'B', 'E')
('A', 'C', 'B')
('A', 'C', 'D')
('A', 'C', 'E')
('A', 'D', 'B')
('A', 'D', 'C')
('A', 'D', 'E')
('A', 'E', 'B')
('A', 'E', 'C')
('A', 'E', 'D')
('B', 'A', 'C')
('B', 'A', 'D')
('B', 'A', 'E')
('B', 'C', 'A')
('B', 'C', 'D')
('B', 'C', 'E')
('B', 'D', 'A')
('B', 'D', 'C')
('B', 'D', 'E')
('B', 'E', 'A')
('B', 'E', 'C')
('B', 'E', 'D')
('C', 'A', 'B')
('C', 'A', 'D')
('C', 'A', 'E')
('C', 'B', 'A')
('C', 'B', 'D')
('C', 'B', 'E')
```

```
('C', 'D', 'A')
('C', 'D', 'B')
('C', 'D', 'E')
('C', 'E', 'A')
('C', 'E', 'B')
('C', 'E', 'D')
('D', 'A', 'B')
('D', 'A', 'C')
('D', 'A', 'E')
('D', 'B', 'A')
('D', 'B', 'C')
('D', 'B', 'E')
('D', 'C', 'A')
('D', 'C', 'B')
('D', 'C', 'E')
('D', 'E', 'A')
('D', 'E', 'B')
('D', 'E', 'C')
('E', 'A', 'B')
('E', 'A', 'C')
('E', 'A', 'D')
('E', 'B', 'A')
('E', 'B', 'C')
('E', 'B', 'D')
('E', 'C', 'A')
('E', 'C', 'B')
('E', 'C', 'D')
('E', 'D', 'A')
('E', 'D', 'B')
('E', 'D', 'C')
('A', 'B', 'C', 'D')
('A', 'B', 'C', 'E')
('A', 'B', 'D', 'C')
('A', 'B', 'D', 'E')
('A', 'B', 'E', 'C')
('A', 'B', 'E', 'D')
('A', 'C', 'B', 'D')
('A', 'C', 'B', 'E')
('A', 'C', 'D', 'B')
('A', 'C', 'D', 'E')
('A', 'C', 'E', 'B')
('A', 'C', 'E', 'D')
('A', 'D', 'B', 'C')
('A', 'D', 'B', 'E')
('A', 'D', 'C', 'B')
('A', 'D', 'C', 'E')
('A', 'D', 'E', 'B')
('A', 'D', 'E', 'C')
('A', 'E', 'B', 'C')
('A', 'E', 'B', 'D')
('A', 'E', 'C', 'B')
('A', 'E', 'C', 'D')
('A', 'E', 'D', 'B')
('A', 'E', 'D', 'C')
('B', 'A', 'C', 'D')
('B', 'A', 'C', 'E')
```

('B', 'A', 'D', 'C')
('B', 'A', 'D', 'E')
('B', 'A', 'E', 'C')
('B', 'A', 'E', 'D')
('B', 'C', 'A', 'D')
('B', 'C', 'A', 'E')
('B', 'C', 'D', 'A')
('B', 'C', 'D', 'E')
('B', 'C', 'E', 'A')
('B', 'C', 'E', 'D')
('B', 'D', 'A', 'C')
('B', 'D', 'A', 'E')
('B', 'D', 'C', 'A')
('B', 'D', 'C', 'E')
('B', 'D', 'E', 'A')
('B', 'D', 'E', 'C')
('B', 'E', 'A', 'C')
('B', 'E', 'A', 'D')
('B', 'E', 'C', 'A')
('B', 'E', 'C', 'D')
('B', 'E', 'D', 'A')
('B', 'E', 'D', 'C')
('C', 'A', 'B', 'D')
('C', 'A', 'B', 'E')
('C', 'A', 'D', 'B')
('C', 'A', 'D', 'E')
('C', 'A', 'E', 'B')
('C', 'A', 'E', 'D')
('C', 'B', 'A', 'D')
('C', 'B', 'A', 'E')
('C', 'B', 'D', 'A')
('C', 'B', 'D', 'E')
('C', 'B', 'E', 'A')
('C', 'B', 'E', 'D')
('C', 'D', 'A', 'B')
('C', 'D', 'A', 'E')
('C', 'D', 'B', 'A')
('C', 'D', 'B', 'E')
('C', 'D', 'E', 'A')
('C', 'D', 'E', 'B')
('C', 'E', 'A', 'B')
('C', 'E', 'A', 'D')
('C', 'E', 'B', 'A')
('C', 'E', 'B', 'D')
('C', 'E', 'D', 'A')
('C', 'E', 'D', 'B')
('D', 'A', 'B', 'C')
('D', 'A', 'B', 'E')
('D', 'A', 'C', 'B')
('D', 'A', 'C', 'E')
('D', 'A', 'E', 'B')
('D', 'A', 'E', 'C')
('D', 'B', 'A', 'C')
('D', 'B', 'A', 'E')
('D', 'B', 'C', 'A')
('D', 'B', 'C', 'E')

```
('D', 'B', 'E', 'A')
('D', 'B', 'E', 'C')
('D', 'C', 'A', 'B')
('D', 'C', 'A', 'E')
('D', 'C', 'B', 'A')
('D', 'C', 'B', 'E')
('D', 'C', 'E', 'A')
('D', 'C', 'E', 'B')
('D', 'E', 'A', 'B')
('D', 'E', 'A', 'C')
('D', 'E', 'B', 'A')
('D', 'E', 'B', 'C')
('D', 'E', 'C', 'A')
('D', 'E', 'C', 'B')
('E', 'A', 'B', 'C')
('E', 'A', 'B', 'D')
('E', 'A', 'C', 'B')
('E', 'A', 'C', 'D')
('E', 'A', 'D', 'B')
('E', 'A', 'D', 'C')
('E', 'B', 'A', 'C')
('E', 'B', 'A', 'D')
('E', 'B', 'C', 'A')
('E', 'B', 'C', 'D')
('E', 'B', 'D', 'A')
('E', 'B', 'D', 'C')
('E', 'C', 'A', 'B')
('E', 'C', 'A', 'D')
('E', 'C', 'B', 'A')
('E', 'C', 'B', 'D')
('E', 'C', 'D', 'A')
('E', 'C', 'D', 'B')
('E', 'D', 'A', 'B')
('E', 'D', 'A', 'C')
('E', 'D', 'B', 'A')
('E', 'D', 'B', 'C')
('E', 'D', 'C', 'A')
('E', 'D', 'C', 'B')
('A', 'B', 'C', 'D', 'E')
('A', 'B', 'C', 'E', 'D')
('A', 'B', 'D', 'C', 'E')
('A', 'B', 'D', 'E', 'C')
('A', 'B', 'E', 'C', 'D')
('A', 'B', 'E', 'D', 'C')
('A', 'C', 'B', 'D', 'E')
('A', 'C', 'B', 'E', 'D')
('A', 'C', 'D', 'B', 'E')
('A', 'C', 'D', 'E', 'B')
('A', 'C', 'E', 'B', 'D')
('A', 'C', 'E', 'D', 'B')
('A', 'D', 'B', 'C', 'E')
('A', 'D', 'B', 'E', 'C')
('A', 'D', 'C', 'B', 'E')
('A', 'D', 'C', 'E', 'B')
('A', 'D', 'E', 'B', 'C')
('A', 'D', 'E', 'C', 'B')
```

('A', 'E', 'B', 'C', 'D')
('A', 'E', 'B', 'D', 'C')
('A', 'E', 'C', 'B', 'D')
('A', 'E', 'C', 'D', 'B')
('A', 'E', 'D', 'B', 'C')
('A', 'E', 'D', 'C', 'B')
('B', 'A', 'C', 'D', 'E')
('B', 'A', 'C', 'E', 'D')
('B', 'A', 'D', 'C', 'E')
('B', 'A', 'D', 'E', 'C')
('B', 'A', 'E', 'C', 'D')
('B', 'A', 'E', 'D', 'C')
('B', 'C', 'A', 'D', 'E')
('B', 'C', 'A', 'E', 'D')
('B', 'C', 'D', 'A', 'E')
('B', 'C', 'D', 'E', 'A')
('B', 'C', 'E', 'A', 'D')
('B', 'C', 'E', 'D', 'A')
('B', 'D', 'A', 'C', 'E')
('B', 'D', 'A', 'E', 'C')
('B', 'D', 'C', 'A', 'E')
('B', 'D', 'C', 'E', 'A')
('B', 'D', 'E', 'A', 'C')
('B', 'D', 'E', 'C', 'A')
('B', 'E', 'A', 'C', 'D')
('B', 'E', 'A', 'D', 'C')
('B', 'E', 'C', 'A', 'D')
('B', 'E', 'C', 'D', 'A')
('B', 'E', 'D', 'A', 'C')
('B', 'E', 'D', 'C', 'A')
('C', 'A', 'B', 'D', 'E')
('C', 'A', 'B', 'E', 'D')
('C', 'A', 'D', 'B', 'E')
('C', 'A', 'D', 'E', 'B')
('C', 'A', 'E', 'B', 'D')
('C', 'A', 'E', 'D', 'B')
('C', 'B', 'A', 'D', 'E')
('C', 'B', 'A', 'E', 'D')
('C', 'B', 'D', 'A', 'E')
('C', 'B', 'D', 'E', 'A')
('C', 'B', 'E', 'A', 'D')
('C', 'B', 'E', 'D', 'A')
('C', 'D', 'A', 'B', 'E')
('C', 'D', 'A', 'E', 'B')
('C', 'D', 'B', 'A', 'E')
('C', 'D', 'B', 'E', 'A')
('C', 'D', 'E', 'A', 'B')
('C', 'D', 'E', 'B', 'A')
('C', 'E', 'A', 'B', 'D')
('C', 'E', 'A', 'D', 'B')
('C', 'E', 'B', 'A', 'D')
('C', 'E', 'B', 'D', 'A')
('C', 'E', 'D', 'A', 'B')
('C', 'E', 'D', 'B', 'A')
('D', 'A', 'B', 'C', 'E')
('D', 'A', 'B', 'E', 'C')

```
('D', 'A', 'C', 'B', 'E')
('D', 'A', 'C', 'E', 'B')
('D', 'A', 'E', 'B', 'C')
('D', 'A', 'E', 'C', 'B')
('D', 'B', 'A', 'C', 'E')
('D', 'B', 'A', 'E', 'C')
('D', 'B', 'C', 'A', 'E')
('D', 'B', 'C', 'E', 'A')
('D', 'B', 'E', 'A', 'C')
('D', 'B', 'E', 'C', 'A')
('D', 'C', 'A', 'B', 'E')
('D', 'C', 'A', 'E', 'B')
('D', 'C', 'B', 'A', 'E')
('D', 'C', 'B', 'E', 'A')
('D', 'C', 'E', 'A', 'B')
('D', 'C', 'E', 'B', 'A')
('D', 'E', 'A', 'B', 'C')
('D', 'E', 'A', 'C', 'B')
('D', 'E', 'B', 'A', 'C')
('D', 'E', 'B', 'C', 'A')
('D', 'E', 'C', 'A', 'B')
('D', 'E', 'C', 'B', 'A')
('E', 'A', 'B', 'C', 'D')
('E', 'A', 'B', 'D', 'C')
('E', 'A', 'C', 'B', 'D')
('E', 'A', 'C', 'D', 'B')
('E', 'A', 'D', 'B', 'C')
('E', 'A', 'D', 'C', 'B')
('E', 'B', 'A', 'C', 'D')
('E', 'B', 'A', 'D', 'C')
('E', 'B', 'C', 'A', 'D')
('E', 'B', 'C', 'D', 'A')
('E', 'B', 'D', 'A', 'C')
('E', 'B', 'D', 'C', 'A')
('E', 'C', 'A', 'B', 'D')
('E', 'C', 'A', 'D', 'B')
('E', 'C', 'B', 'A', 'D')
('E', 'C', 'B', 'D', 'A')
('E', 'C', 'D', 'A', 'B')
('E', 'C', 'D', 'B', 'A')
('E', 'D', 'A', 'B', 'C')
('E', 'D', 'A', 'C', 'B')
('E', 'D', 'B', 'A', 'C')
('E', 'D', 'B', 'C', 'A')
('E', 'D', 'C', 'A', 'B')
('E', 'D', 'C', 'B', 'A')

Combinations:
('A',)
('B',)
('C',)
('D',)
('E',)
('A', 'B')
('A', 'C')
('A', 'D')
```

```
('A', 'E')
('B', 'C')
('B', 'D')
('B', 'E')
('C', 'D')
('C', 'E')
('D', 'E')
('A', 'B', 'C')
('A', 'B', 'D')
('A', 'B', 'E')
('A', 'C', 'D')
('A', 'C', 'E')
('A', 'D', 'E')
('B', 'C', 'D')
('B', 'C', 'E')
('B', 'D', 'E')
('C', 'D', 'E')
('A', 'B', 'C', 'D')
('A', 'B', 'C', 'E')
('A', 'B', 'D', 'E')
('A', 'C', 'D', 'E')
('B', 'C', 'D', 'E')
('A', 'B', 'C', 'D', 'E')
```

In [ ]: