1. **Python Program to Remove empty List from List.**

```
# Initializing list
test_list = [5, 6, [], 3, [], [], 9]

# printing original list
print("The original list is : " + str(test_list))

# Remove empty List from List
# using list comprehension
res = [ele for ele in test_list if ele != []]

# printing result
print("List after empty list removal : " + str(res))
```

2. **Python program for Simple Calculator.**

```
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))
operation = input("Enter the operation (+, -, *, /): ")

result = None
if operation == "+":
   result = num1 + num2
elif operation == "-":
   result = num1 - num2
elif operation == "*":
   result = num1 * num2
elif operation == "/":
   result = num1 / num2

print(f"Result: {result}")
```

3. **Sort the values of first list using second list in Python**

```
# Python program to sort one list using the other list

def sort_list(list1, list2):

        zipped_pairs = zip(list2, list1)

        z = [x for _, x in sorted(zipped_pairs)]

        return z

# driver code
```

```
x = ["a", "b", "c", "d", "e", "f", "g", "h", "i"]
y = [0, 1, 1, 0, 1, 2, 2, 0, 1]

print(sort_list(x, y))

x = ["g", "e", "e", "k", "s", "f", "o", "r", "g", "e", "e", "k", "s"]
y = [0, 1, 1, 0, 1, 2, 2, 0, 1]

print(sort_list(x, y))
```

Output:

```
['a', 'd', 'h', 'b', 'c', 'e', 'i', 'f', 'g']
['g', 'k', 'r', 'e', 'e', 'g', 's', 'f', 'o']
```

4. Transpose Matrix in Single Line using List Comprehension

```
m = [[1, 2], [3, 4], [5, 6]]
for row in m:
  print(row)
rez = [[m[j][i] for j in range(len(m))] for i in range(len(m[0]))]
print("\n")
for row in rez:
  print(row)
```

Output:

```
[1, 2]
[3, 4]
[5, 6]
[1, 3, 5]
[2, 4, 6]
```

5. Print anagrams together in Python using List and Dictionary.

```
def allAnagram(input):

    dict = {}

    # traverse list of strings
    for strVal in input:


        key = ''.join(sorted(strVal))

        if key in dict.keys():
            dict[key].append(strVal)
        else:
            dict[key] = []
            dict[key].append(strVal)
```

```
    # traverse dictionary and concatenate values of keys together
    output = ""
    for key,value in dict.items():
        output = output + ' '.join(value) + ' '

    return output

# Driver function
if __name__ == "__main__":
    input=['cat', 'dog', 'tac', 'god', 'act']
    print (allAnagram(input))
```

6. Python program to Order Tuples Using lists and index() method.

```
# Python3 code to demonstrate working of
# Order Tuples by List

# initializing list
test_list = [('Gfg', 3), ('best', 9), ('CS', 10), ('Geeks', 2)]

# printing original list
print("The original list is : " + str(test_list))

# initializing order list
ord_list = ['Geeks', 'best', 'CS', 'Gfg']

res=[]
x=[]
for i in test_list:
    x.append(i[0])
for i in ord_list:
    if i in x:
        res.append(test_list[x.index(i)])
# printing result
print("The ordered tuple list : " + str(res))
```

7. Write a Python script for a "Guess the Number" game.

```
import random

def guess_the_number():
    target_number = random.randint(1, 100)
    attempts = 0

    while True:
        user_guess = int(input("Guess the number (between 1 and 100):
"))
        attempts += 1
```

```
        if user_guess == target_number:
            print(f"Congratulations! You guessed the number
{target_number} in {attempts} attempts.")
            break
        elif user_guess = target_number:
            print("Too low. Try again.")
        else:
            print("Too high. Try again.")

if _name_ == "_main_":
    print("Welcome to the Guess the Number game!")
    guess_the_number()
```

8. Comparing arrays

This problem helps one to understand the key concepts of an array(list) in Python. Two arrays are said to be the same if they contain the same elements and in the same order. However, in this problem, we will compare two arrays to see if they are same, but with a slight twist. Here, two arrays are the same if the elements of one array are squares of elements of other arrays and regardless of the order. Consider two arrays **a** and **b**.

a = `[121, 144, 19, 4]`

`b = [16,121,361, 20736]`

Here **b** can be written as:
b = [4*4,11*11,19*19, 144*144]

which is a square of every element of **a**. Hence, they are same. If **a** or **b** are None, our program should written False.

```
# function to compare the arrays
def comp(array1, array2):

    # checking if any array is None
    if array1 is None or array2 is None:
        return False

    # checking if any of the array
    # is a square of the other
    if (sorted(array1) == sorted([i ** 2 for i in array2])) or (sorted(array2)
== sorted([i ** 2 for i in array1])):
        return True

    return False

# Driver Code
comp([1,2,3,4], [1,4,9,16])
```

9. Direction Catastrophe

```
opposite = {'NORTH': 'SOUTH',
                    'EAST': 'WEST',
                    'SOUTH': 'NORTH',
                    'WEST': 'EAST'}
# Function to find the reduced
# direction
def dirReduc(givenDirections):
        finalDirections = []
        for d in range(0, len(givenDirections)):
                if finalDirections:
                        if finalDirections[-1] == opposite[givenDirections[d]]:
                                finalDirections.pop()
                        else:
                                finalDirections.append(givenDirections[d])
                else:
                        finalDirections.append(givenDirections[d])
        return finalDirections


# Driver Code
print(dirReduc(["NORTH", "SOUTH", "SOUTH", "EAST", "WEST", "NORTH", "WEST"]))
```

10. Write a python code to accept a string and count the number of vowels and consonants.

```
inp= input('Enter the string : ')
v=0
c=0
for i in inp:
  if i in ['a','e','i','o','u']:
    v=v+1
  elif i == " ":
    pass
  else :
    c=c+1
print( 'The number of vowels is ', v)
print( 'The number of components is ', c)
```