1. Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing', add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged.

```
def is_number(num):
    try:
        float(num)
        return True
    except ValueError:
        return False

def modify_string(input_str):
    if is_number(input_str) == False:
        if len(input_str) < 3:
            return input_str
        elif input_str[-3:] == 'ing':
            return input_str + 'ly'
        else:
            return input_str + 'ing'
    else:
        return False

InputValue = input("Enter a string: ")
result = modify_string(InputValue)
if result == False:
    print("Entered input is not String")
else:
    print("Modified string:", result)
```

2. Write a Python function that takes a list of words and return the longest word and the length of the longest one.

```
def longest_word_prog(words):
    if not words:
        return None, 0

    longest_word = words[0]
    max_length = len(longest_word)

    for word in words[1:]:
        current_length = len(word)
        if current_length > max_length:
            longest_word = word
            max_length = current_length

    return longest_word, max_length

input_string = input("Enter a list of strings separated by space: ")
string_array = input_string.split()
longest_word, length = longest_word_prog(string_array)

print("Longest word:", longest_word)
print("Length of the longest word:", length)
```

3. Write a Python program to pack consecutive duplicates of a given list of elements into sublists.

```
def pack_duplicates(list):
    packed_list = []
    current_group = []

    for element in list:
        if not current_group or element == current_group[-1]:
            current_group.append(element)
        else:
            packed_list.append(current_group)
            current_group = [element]

    if current_group:
        packed_list.append(current_group)

    return packed_list

# Example usage:
input_string = input("Enter a list of strings separated by space: ")
string_array = input_string.split()
result = pack_duplicates(string_array)


print("Original list:", string_array)
print("Packed list:", result)
```

4.Write a Python program to find the item with the most occurrences in a given list.

```
def most_common_item(input_list):
    if not input_list:
        return None

    item_counts = {}
    for item in input_list:
        if item in item_counts:
            item_counts[item] += 1
        else:
            item_counts[item] = 1

    most_common = max(item_counts, key=item_counts.get)

    return most_common,


string_array = input("Enter a list of strings separated by space: ")
string_array = string_array.split()
result = most_common_item(string_array)

print("List:", string_array)
print("Most common item:", result)
```

5.Write a Python program to find the highest 3 values of corresponding keys in a dictionary.

```
def highest_values(dictionary):
    sorted_items = sorted(dictionary.items(), key=lambda x: x[1], reverse=True)

    top_three = sorted_items[:3]

    return top_three

sample_dict = {'a': 101, 'b': 555, 'c': 81, 'd': 1112, 'e': 3}
result = highest_values(sample_dict)

print("Original dictionary:", sample_dict)
print("Top 3 values and keys:", result)
```

6. Write a Python program to get the top three items in a shop.
Sample data: {'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55, 'item5': 24}
Expected Output:

```
item4 55
item1 45.5
item3 41.3
```

```

def Sortedtop_three(dictionary):
    sorted_items = sorted(dictionary.items(), key=lambda x: x[1], reverse=True)

    top_three = sorted_items[:3]

    return top_three


Sample_data = {'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55, 'item5': 24}

sorted_top_three = Sortedtop_three(Sample_data)
sorted_top_three_dict = dict(sorted_top_three)

for key, value in sorted_top_three_dict.items():
    print(f"Key: {key}, Value: {value}")
```