Assignment - 2

V V Uday Bhaskar HTno: 2306DSP107

Question 1 of 5

Given an integer, perform the following condition actions:

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.

Solution 1

In [1]:
```
string = input()
if len(string) >= 3:
    string += 'ly' if string.endswith('ing') else 'ing'
print(string)
```

```
loving
lovingly
```

Solution 2

In [3]:
```
# Define a function named add_string that takes one argument, 'str1'.
def add_string(str1):
    # Get the length of the input string 'str1' and store it in the variable 'length'.
    length = len(str1)

    # Check if the length of 'str1' is greater than 2 characters.
    if length > 2:
        # If the last three characters of 'str1' are 'ing', add 'ly' to the end.
        if str1[-3:] == 'ing':
            str1 += 'ly'
        else:
            # If the last three characters are not 'ing', add 'ing' to the end.
            str1 += 'ing'

    # Return the modified 'str1'.
    return str1
```

Question 2 of 5

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

In [6]:
```
L = input().split(',')
max=0
m=''
for i in range(len(L)):
    if len(L[i])>=max:
```

```
        max=len(L[i])
        m=''
        m+=L[i]
print(m)
```

```
A = [ant, Ball, Duck, Elephant]
 Elephant]
```

Method 2

In [9]:
```
def first_3(L) :
        fmax, smax, tmax = L[0], L[1], L[2]

        for x in L :
                if x > fmax :
                        tmax = smax
                        smax = fmax
                        fmax = x
                elif x > smax and x < fmax :
                        tmax = smax
                        smax = x
                elif x > tmax and x < smax :
                        tmax = x
        return fmax, smax, tmax
```

Question 3 of 5

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

Method 1

In [10]:
```
class Solution:
   def solve(self, nums):
      if not nums:
         return []
      result = [[nums[0]]]
      j = 0
      for i in range(1, len(nums)):
         if nums[i] != nums[i - 1]:
            result.append([])
            j += 1
            result[j].append(nums[i])
      return result
ob = Solution()
nums = [5, 5, 2, 7, 7, 7, 2, 2, 2, 2]
print(ob.solve(nums))
```

```
[[5], [2], [7], [2]]
```

Question 4 of 5

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

In [13]:
```
# Define a function 'max_occurrences' that finds the item with the maximum occurrences
def max_occurrences(nums):
    # Initialize 'max_val' to 0 and 'result' to the first item in the list
    max_val = 0
    result = nums[0]
```

```
    # Iterate through the elements in the 'nums' list
    for i in nums:
        # Count the occurrences of the current element 'i' in the list
        occu = nums.count(i)

        # Check if the number of occurrences is greater than the current maximum
        if occu > max_val:
            # Update 'max_val' and 'result' if a higher occurrence is found
            max_val = occu
            result = i

    # Return the item with the maximum occurrences
    return result

# Create a list 'nums' with integers
nums = [2,3,8,4,7,9,8,2,6,5,1,6,1,2,3,4,6,9,1,2]

# Print a message indicating the original list
print ("Original list:")
# Print the contents of the 'nums' list
print(nums)

# Print a message indicating the item with the maximum occurrences in the list
print("\nItem with maximum occurrences of the said list:")
# Call the 'max_occurrences' function with 'nums' and print the result
print(max_occurrences(nums))
```

```
Original list:
[2, 3, 8, 4, 7, 9, 8, 2, 6, 5, 1, 6, 1, 2, 3, 4, 6, 9, 1, 2]

Item with maximum occurrences of the said list:
2
```

Question 5 of 5

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

In [14]:
```
dict = {'A' : 25, 'B' : 55, 'C': 15, 'D': 85, 'E' : 20, 'F' : 63, 'G' : 42 }
high = sorted(dict, key = dict.get, reverse = True)
for index in range(3) :
        print(high[index])
```

```
D
F
B
```

Thankyou