```
#1) Write a Python program which takes a list and returns a list with the
#elements
#"shifted left by one position" so [1, 2, 3] yields [2, 3, 1].
#Example: [1, 2, 3] → [2, 3, 1]    & [11, 12, 13] → [12, 13, 11]

print("Shift List Elements Left by One Position")

# Input list from user
elements = input("Enter list elements separated by spaces: ").split()

# Convert strings to integers
for i in range(len(elements)):
    elements[i] = int(elements[i])

# Shift left by one position
shifted = elements[1:] + elements[:1]

# Display result
print("Shifted List:", shifted)
```

```
Shift List Elements Left by One Position
Enter list elements separated by spaces: 1 2 3
Shifted List: [2, 3, 1]
```

```
#2) Consider the list lst=[9,8,7,6,5,4,3]. Write the Python program which perrforms the following operation.
#A. Insert element 10 at beginning of the list.
#B. Insert element 2 at end of the list.
#C. Delete the element at index position 5.
#D. Print all elements in reverse order

lst=[9,8,7,6,5,4,3]
print("Original list:", lst)

# A. Insert element 10 at beginning of the list
lst.insert(0, 10)
print("After inserting 10 at beginning:", lst)

# B. Insert element 2 at end of the list
lst.append(2)
print("After inserting 2 at end:", lst)

# C. Delete the element at index position 5
del lst[5]
print("After deleting element at index 5:", lst)

# D. Print all elements in reverse order
lst.reverse()
print("List in reverse order:", lst)
```

```
Original list: [9, 8, 7, 6, 5, 4, 3]
After inserting 10 at beginning: [10, 9, 8, 7, 6, 5, 4, 3]
After inserting 2 at end: [10, 9, 8, 7, 6, 5, 4, 3, 2]
After deleting element at index 5: [10, 9, 8, 7, 6, 4, 3, 2]
List in reverse order: [2, 3, 4, 6, 7, 8, 9, 10]
```

```
#3) Write a Python program which will return the sum of the numbers in the
#array, returning 0 for an empty array. Except the number 13 is very unlucky,
#so it does not count and number that come immediately after 13 also do not count.

numbers=input("Enter the numbers giving spaces ").split()

for i in range(len(numbers)):
  numbers[i]=int(numbers[i])

tot=0
i=0

while i<len(numbers):
  if numbers[i]==13:
    i=i+2
  else:
    tot=tot+numbers[i]
    i+=1

print("The sum is: ", tot)
```

```
Enter the numbers giving spaces 12 14 14 13 2
The sum is:  40
```

```
#4) Write a Python program to Check Whether a String is Palindrome or Not

print("Check Whether a String is Palindrome or Not")

text = input("Enter a string: ")

# Remove spaces and convert to lowercase for accurate comparison (optional)
clean_text = text.replace(" ", "").lower()

# Reverse the string
reversed_text = clean_text[::-1]

# Compare original and reversed
if clean_text == reversed_text:
    print("The string is a palindrome.")
else:
    print("The string is not a palindrome.")
```

```
Check Whether a String is Palindrome or Not
Enter a string: ana
The string is a palindrome.
```

```
#5) Write a python program to do the following:
#A. To sum all the items in a list.
#B. To multiplies all the items in a list
#C. To get the largest number from a list.
#D. To get the smallest number from a list.
#E. To remove duplicates from a list.
#F. To check a list is empty or not
#G. To select an item randomly from a list.
#H. To clone or copy a list
#I.  To find the second smallest number in a list.
#J. To find the second largest number in a list
#K. To get unique values from a list.
#L. To remove the K'th element from a given list, print the new list.
#M. To insert an element at a specified position into a given list.

import random

print("Working with Lists in Python")

# Sample list
lst = [10, 20]

# A. Sum all items in a list
sum_items = 0
for item in lst:
    sum_items += item
print("A. Sum of all items:", sum_items)

# B. Multiply all items in a list
product = 1
for item in lst:
    product *= item
print("B. Product of all items:", product)

# C. Get the largest number
largest = lst[0]
for item in lst:
    if item > largest:
        largest = item
print("C. Largest number:", largest)

# D. Get the smallest number
smallest = lst[0]
for item in lst:
    if item < smallest:
        smallest = item
print("D. Smallest number:", smallest)

# E. Remove duplicates
no_duplicates = []
for item in lst:
    if item not in no_duplicates:
        no_duplicates.append(item)
print("E. List without duplicates:", no_duplicates)
```

```
# F. Check if list is empty
if len(lst) == 0:
    print("F. The list is empty")
else:
    print("F. The list is not empty")

# G. Select an item randomly
random_item = random.choice(lst)
print("G. Randomly selected item:", random_item)

# H. Clone or copy a list
copy_list = lst[:]
print("H. Cloned list:", copy_list)

# I. Second smallest number
unique_sorted = sorted(list(set(lst)))
if len(unique_sorted) >= 2:
    print("I. Second smallest number:", unique_sorted[1])
else:
    print("I. Not enough unique elements")

# J. Second largest number
if len(unique_sorted) >= 2:
    print("J. Second largest number:", unique_sorted[-2])
else:
    print("J. Not enough unique elements")

# K. Get unique values
unique_values = list(set(lst))
print("K. Unique values in list:", unique_values)

# L. Remove K'th element (e.g., K = 3, remove element at index 3)
k = 3
if 0 <= k < len(lst):
    new_list = lst[:k] + lst[k+1:]
    print("L. List after removing element at index", k, ":", new_list)
else:
    print("L. Invalid index")

# M. Insert an element at a specified position
element = 99
position = 2
if 0 <= position <= len(lst):
    lst.insert(position, element)
    print("M. List after inserting", element, "at position", position, ":", lst)
else:
    print("M. Invalid position")
```

```
Working with Lists in Python
A. Sum of all items: 30
B. Product of all items: 200
C. Largest number: 20
D. Smallest number: 10
E. List without duplicates: [10, 20]
F. The list is not empty
G. Randomly selected item: 10
H. Cloned list: [10, 20]
I. Second smallest number: 20
J. Second largest number: 10
K. Unique values in list: [10, 20]
L. Invalid index
M. List after inserting 99 at position 2 : [10, 20, 99]
```

```
#6) Write a python program to show the use of count and index method in  tuples.

# Define a tuple
m_tuple = (10, 20, 30, 20, 40, 50, 20, 60)

print("Tuple:", m_tuple)

# Use count() method
# Counts how many times 20 appears in the tuple
count_20 = m_tuple.count(20)
print("Count of 20 in the tuple:", count_20)

# Use index() method
# Finds the index of the first occurrence of 30
index_30 = m_tuple.index(30)
```

```
print("Index of first occurrence of 30:", index_30)

# Another example: first occurrence of 20
index_20 = m_tuple.index(20)
print("Index of first occurrence of 20:", index_20)
```

```
Tuple: (10, 20, 30, 20, 40, 50, 20, 60)
Count of 20 in the tuple: 3
Index of first occurrence of 30: 2
Index of first occurrence of 20: 1
```