Software Development – List, Tuple

Python programming exercises

1. Write a program that finds the maximum value of the given list, assuming that the list contains at least one element.

Try your program with the following array

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 4 | 7 | 4 | 23 | 5 | 1 | 4 | 8 | 9 |

def find\_maximum(lst):

    if len(lst) > 0:

        maximum\_value = max(lst)

        print(f"The maximum value in the list is: {maximum\_value}")

    else:

        print("The list is empty.")

# Example usage

my\_list = [3, 7, 1, 9, 4,10,11,20]

find\_maximum(my\_list)

def Max():

     ary=[2,4,7,4,23,5,1,4,8,9]

     a=max(ary)

     print(a)

Max()

1. Write a program that calculates the average value of the given list. Try your program with the following list

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 7 | 1 | 5 | 11 | 53 | 12 | 46 | 84 | 23 |

arx2=[4,7,1,5,11,53,12,46,84,23]

def average(arrx,n):

    i=0

    sum=0

    while i<n:

        sum+=arrx[i]

        i+=1

    print(sum)

    print(n)

    av\_num=float(sum/n)

    print('The average is',av\_num)

average(arx2,len(arx2))

import statistics

def average():

    arx2=[4,7,1,5,11,53,12,46,84,23]

    print(statistics.mean([4,7,1,5,11,53,12,46,84,23] ))

average()

1. Write a program that prints the given list of integers in reverse order. Try your program with the following list

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 6 | 7 | 45 | 23 | 53 | 14 | 45 | 89 | 5 |
|  |  |  |  |  |  |  |  |  |  |

list1=[2,6,7,45,23,53,14,45,89,5]

def revers():

     list1.sort(reverse=1)

     print(list1)

revers()

1. Write a program that accepts two lists of integers and prints true if each element in the first list is less than the element at the same index in the second list. Your program should print false if the lists are not the same length.

def compare\_lists(list1, list2):

    if len(list1) != len(list2):

        print("False (Lists are not the same length)")

        return

    result = all(element1 < element2 for element1, element2 in zip(list1, list2))

    print(result)

# Example usage

list1 = [1, 2, 3, 4, 5]

list2 = [6, 7, 8, 9, 8]

compare\_lists(list1, list2)

def compare():

#creating list 1

    l1=[]

    print('\*\*Enter 00 to mark end of list\*\*')

    x=int(input('Enter values for list 1 '))

    l1.append(x)

    while x!=00:

        x=int(input('Enter values for list 1 '))

        l1.append(x)

        if x==00:

            break

#creating list 2

    l2=[]

    print('\*\*Enter 00 to mark end of list\*\*')

    y=int(input('Enter values for list 2 '))

    l2.append(y)

    while y!=00:

        y=int(input('Enter values for list 2 '))

        l2.append(y)

        if y==00:

            break

    print('\n')

    print('List1:',l1)

    print('List2:',l2)

#comparing the lists

    if len(l1)!=len(l2):

        print('False''\nLists must be of the same size')

    else:

        for i in range(0,len(l1)):

            if l1[i]<l2[i]:

                print('True')

                print(l1[i],'Is less than',l2[i])

compare()

1. Write a program that accepts a list of integers and two indexes and swaps the elements at those indexes

def swap\_elements(lst, index1, index2):

    if 0 <= index1 < len(lst) and 0 <= index2 < len(lst):

        # Swap elements at index1 and index2

        lst[index1], lst[index2] = lst[index2], lst[index1]

        print(f"Swapped elements at indices {index1} and {index2}: {lst}")

    else:

        print("Invalid indices. Please provide valid indices within the list range.")

# Example usage

my\_list = [1, 2, 3, 4, 5]

# Swap elements at index 1 and index 3

swap\_elements(my\_list, 1, 3)

#5 Swapping 2 elements using index of a list

n=int(input('Enter number of elements: '))#number of elements

list3=[]

def ls3(list3,m):

        for num in range(0,m):

            num=int(input(f'Enter number {num}: '))

            list3.append(num)

        print('Before',list3)

    #ls3(list3,m)

print('\n')

def swap(list3):#function to swap

    x=int(input('Enter position1 to swap: '))

    y=int(input('Enter position2 to swap: '))

    i=list3[y]

    list3[y]=list3[x]

    list3[x]=i

    print('After',list3)

ls3(list3,n)

swap(list3)

def swap(list3,m):#function to swap

    for num in range(0,m):

        num=int(input(f'Enter number {num}: '))

        list3.append(num)

    print('Before',list3)

    x=int(input('Enter position1 to swap: '))

    y=int(input('Enter position2 to swap: '))

    # i=list3[y]

    # list3[y]=list3[x]

    # list3[x]=i

    list3[x],list3[y]=list3[y],list3[x]#swap

    print('After',list3)

#ls3(list3,n)

swap(list3,n)

1. Write a program that accepts two lists of integers and prints a new list containing all elements of the first list followed by all elements of the second.

def concat(list1, list2):

    result = list1 + list2

    print(result)

# Example usage

list1 = [7, 2, 3, 4, 5]

list2 = [6, 7, 8, 9]

concat(list1, list2)

1. Write a program that accepts a list of integers and an integer value as its parameters and prints the last index at which the value occurs in the list. The program should print –1 if the value is not found. For example, in the list [74, 85, 102, 99, 101, 85, 56], the last index of the

value 85 is 5.

def last\_index\_of\_value(lst, value):

    try:

        last\_index = len(lst) - 1 - lst[::-1].index(value)

        print(f"The last index of the value {value} is: {last\_index}")

    except ValueError:

        print(f"The value {value} is not found in the list. Last index: -1")

# Example usage

my\_list = [74, 85, 102, 99, 101, 85, 56]

search\_value = 85

last\_index\_of\_value(my\_list, search\_value)

1. Write a program that prints the range of values in a list of integers. The range is defined as 1 more than the difference between the maximum and minimum values in the list. For example, if a list contains the values [36, 12, 25, 19, 46, 31, 22], the program should return

35. You may assume that the list has at least one element.

def calculate\_range(lst):

    min\_value = min(lst)

    max\_value = max(lst)

    data\_range = max\_value - min\_value + 1

print(f"The range of values in the list is: {data\_range}")

# Example usage

my\_list = [36, 12, 25, 19, 46, 31, 22]

calculate\_range(my\_list)

1. Write a program that accepts a list of integers, a minimum value, and a maximum value and prints the count of how many elements from the list fall between the minimum and maximum (inclusive). For example, in the list [14, 1, 22, 17, 36, 7, -43, 5], for minimum value 4 and maximum value 17, there are four elements whose values fall between 4 and 17.
2. Write a program that accepts a list of real numbers and prints true if the list is in sorted (nondecreasing) order and false otherwise. For example, if lists named list1 and list2 store [16.1, 12.3, 22.2, 14.4] and [1.5, 4.3, 7.0, 19.5, 25.1, 46.2] respectively, the program should print false for list1 and true for list2 respectively. Assume the list has at least one element. A one-element list is sorted.