CHAPTER -4 LISTS

Contents

[1. Program to find sum of elements in a list 1](#_Toc510130557)

[2. Program to find sum of elements at even position in a list 2](#_Toc510130558)

[3. Function that takes a list as input and returns sublist of all the elements of the list that are less than entered number 2](#_Toc510130559)

[4. Find the largest element in a list and also return at which position the element is found and after how many comparisons 3](#_Toc510130560)

[5. Find the number of even and odd elements in the list 3](#_Toc510130561)

[6. Program to remove duplicate elements in a list 4](#_Toc510130562)

[7. Program to search specific element in a list 4](#_Toc510130563)

[8. Program to sort elements in a list 5](#_Toc510130564)

[9. Write a function that takes two lists and return True if there is atleast one common element in both lists else False 5](#_Toc510130565)

[10.Write a function that returns true if entered number is dudeney number, otherwise false. 6](#_Toc510130566)

# Program to find sum of elements in a list

li = [1,2,3,4,5,6,7,8]

s = 0

for x in li:

s = s+x

print(s)

Output : 36

# Program to find sum of elements at even position in a list

li = [1,2,3,4,5,6,7,8]

s = 0

for x in li[::2]:

s = s+x

print(s)

Output : 16

# Function that takes a list as input and returns sublist of all the elements of the list that are less than entered number

def func():

mylist = raw\_input('Enter your list: ')

mylist = [int(x) for x in mylist.split(',')]

newlist = []

a = int(raw\_input("enter a number "))

for num in mylist:

if a > num:

newlist.append(num)

print(newlist)

func()

Output : Enter your list: 4,5,6,7,8,1,2

enter a number 4

[1, 2]

# Find the largest element in a list and also return at which position the element is found and after how many comparisons

li = [1,2,3,55,45,6]

max = 0

count = 0

for i in li:

if i > max:

max = i

count += 1

print("This is the largest number : %d" %max)

print("It is found at position number %d" % li.index(max))

print("It is found after %d comparisons" %count)

Output : This is the largest number : 55

It is found at position number 3

It is found after 4 comparisons

# Find the number of even and odd elements in the list

li = [1,2,3,55,45,6]

even = 0

odd = 0

for element in li:

if element % 2==0:

even += 1

else:

odd += 1

print("The number of even elements are : %d" %even )

print ("The number of odd elements are : %d" %odd)

Output : The number of even elements are : 2

The number of odd elements are : 4

# Program to remove duplicate elements in a list

li = [1,2,3,4,3,3,88,8,9,9,8,6]

new = []

for x in li:

if x not in new:

new.append(x)

print(new)

Output : [1, 2, 3, 4, 88, 8, 9, 6]

# Program to search specific element in a list

def linearSearch():

mylist = raw\_input('Enter your list: ')

mylist = [int(x) for x in mylist.split(',')]

item = int(raw\_input("Enter the number to be searched."))

found = False

position = 0

while position < len(mylist) and not found:

if mylist[position] == item:

found = True

position = position + 1

if found is True:

print("Yes this element exits.")

elif found is False:

print("This element doesn't exist")

linearSearch()

Output: Enter your list: 2,3,4,5,6,77,8,9

Enter the number to be searched.34

This element doesn't exist

# Program to sort elements in a list

mylist = raw\_input('Enter your list: ')

mylist = [int(x) for x in mylist.split(',')]

mylist.sort()

print(mylist)

Output: Enter your list: 8,9,5,4,46,32,98

[4, 5, 8, 9, 32, 46, 98]

# Write a function that takes two lists and return True if there is atleast one common element in both lists else False

def func(list1,list2):

lis = []

for a in list1:

if a in list2:

lis.append(a)

f lis !=0:

return True

else:

return False

func([1,2,3,4],[2,5,6,7])

Output: True

# Write a function that returns true if entered number is dudeney number, otherwise false.

def check\_num(number):

add = 0

for i in str(number)[::-1]:

digit = int(i)

add = add + digit

check = add\*\*3

if (check == number):

return True

else:

return False

check\_num(512)

Output : True