list1 = [1,2,3,5,8,2,5.2]

total = 0

i = 0

**while** i < len(list1):

total = total + list1[i]

i = i + 1

average = total / len(list1)

**print** average

myList = [1,2,3,5,8,2,5.2]

i = 0

**while** i < len(myList):

**print** myList[i]

i = i + 1

x=[1,3,45,23,**"ram"**,56]  
search=int(input(**"which element you want to delete?"**))  
**if** search **not in** x:  
 print(**"search item not found"**)  
**else**:  
 x.remove(search)  
print(**"yourr list is"**,x)

Python | Remove all occurrences a given element from the list

Here, we are going to implement a **Python program that will remove all occurrences a given element from the list**.  
Submitted by **IncludeHelp**, on August 06, 2018

**Given a list, and we have to remove all occurrences of a given element from the list in Python.**

**Example:**

Input:

list = [10, 20, 10, 30, 10, 40, 10, 50]

n = 10

Output:

list after removing 10 from the list

list = [20, 30, 40, 50]

**Logic:**

* Run a while loop from 0th element to last element's index.
* Check the element whether it is equal to the number (which is to be removed) or not.
* If any element of the list is equal to the number (which is to be removed), remove that element from the list.
* To remove the number from the list, use [list.remove() method](https://www.includehelp.com/python/remove-first-occurrence-of-a-given-element-in-the-list.aspx).
* After removing the number/element from the list, decrease the length, because one item is deleted, and then continue the loop to check the next item at same index (because after removing the element – next elements shifts to the previous index.
* If element is not found (i.e. is not removed), then increase the loop counter to check next element.

**Example:**

# list with integer elements

list = [10, 20, 10, 30, 10, 40, 10, 50]

# number (n) to be removed

n = 10

# print original list

**print** ("Original list:")

**print** (list)

# loop to traverse each element in list

# and, remove elements

# which are equals to n

i=0 #loop counter

length = len(list) #list length

**while**(i<length):

**if**(list[i]==n):

list.remove (list[i])

# as an element is removed

# so decrease the length by 1

length = length -1

# run loop again to check element

# at same index, when item removed

# next item will shift to the left

**continue**

i = i+1

# print list after removing given element

**print** ("list after removing elements:")

**print** (list)

**Output**

Original list:

[10, 20, 10, 30, 10, 40, 10, 50]

list after removing elements:

[20, 30, 40, 50]

# Python | Program to sort the elements of given list in Ascending and Descending Order

Here, we are going to learn how to **sort the elements of a list in Ascending and Descending order in Python**?  
Submitted by **IncludeHelp**, on July 24, 2018

**Given a list of the elements and we have to sort the list in Ascending and the Descending order in Python.**

## **Python list.sort() Method**

sort() is a inbuilt method in Python, it is used to sort the elements/objects of the list in Ascending and Descending Order.

**Sorting elements in Ascending Order (list.sort())**

**Syntax:**

list.sort()

**Program to sort list elements in Ascending Order**

# List of integers

num = [10, 30, 40, 20, 50]

# sorting and printing

num.sort()

**print** (num)

# List of float numbers

fnum = [10.23, 10.12, 20.45, 11.00, 0.1]

# sorting and printing

fnum.sort()

**print** (fnum)

# List of strings

str = ["Banana", "Cat", "Apple", "Dog", "Fish"]

# sorting and printing

str.sort()

**print** (str)

**Output**

[10, 20, 30, 40, 50]

[0.1, 10.12, 10.23, 11.0, 20.45]

['Apple', 'Banana', 'Cat', 'Dog', 'Fish']

**Sorting in Descending Order (list.sort(reverse=True))**

To sort a list in descending order, we pass reverse=True as an argument with sort() method.

**Syntax:**

list.sort(reverse=True)

**Program to sort list elements in Descending Order**

# List of integers

num = [10, 30, 40, 20, 50]

# sorting and printing

num.sort(reverse=True)

**print** (num)

# List of float numbers

fnum = [10.23, 10.12, 20.45, 11.00, 0.1]

# sorting and printing

fnum.sort(reverse=True)

**print** (fnum)

# List of strings

str = ["Banana", "Cat", "Apple", "Dog", "Fish"]

# sorting and printing

str.sort(reverse=True)

**print** (str)

**Output**

[50, 40, 30, 20, 10]

[20.45, 11.0, 10.23, 10.12, 0.1]

['Fish', 'Dog', 'Cat', 'Banana', 'Apple']

Python | Program to find the position of minimum and maximum elements of a list

Here, we are going to learn how to **find and print the position/index of minimum and maximum elements of a list**? To find the minimum and maximum elements, we use **min() and max() methods in Python**.  
Submitted by **IncludeHelp**, on July 25, 2018

**Given a list and we have to find the index/position of minimum and maximum elements of a list in Python.**

**Prerequisite:**

* [Python | min() Method](https://www.includehelp.com/python/find-minimum-value-from-given-parameters-using-min.aspx)
* [Python | max() Method](https://www.includehelp.com/python/find-maximum-value-from-given-parameters-using-max.aspx)

**Example:**

Input:

list = [10, 1, 2, 20, 3, 20]

Output:

Positive of minimum element: 1

Positive of maximum element: 3

**Logic:**

To **find the positions/indexes of minimum and maximum elements of a list**, we need to find the maximum and minimum elements of the list – to find the maximum element of the list, we will use max(list) and to find the minimum element of the list, we will use min(list).

And, to get their indexes, we will use list.index(max(list)) and list.index(min(list)).

**Program to find the position of min and max elements of a list in Python**

# declare a list of Integers

list = [10, 1, 2, 20, 3, 20]

# min element's position/index

min = list.index (min(list))

# max element's position/index

max = list.index (max(list))

# printing the position/index of min and max elements

**print** "position of minimum element: ", min

**print** "position of maximum element: ", max

**Output**

position of minimum element: 1

position of maximum element: 3

**Explanation:**

* The minimum number of the list is 1 and it is at 1st position in the list. To get it’s index, we use list.index(min(list)) statement, min(list) returns 1 (as minimum element) and list.index(1) returns the index of 1 from the list. Hence, the position of minimum element is: 1
* The maximum number of the list if 20 and it is two times in the list, first occurrence of 20 is at 3rdposition and the second occurrence of 20 is at 5th position. Statement max(list) returns the maximum element of the list, which is 20 and the statement list.index(20) returns the [index/position of first matched element](https://www.includehelp.com/python/print-the-index-of-first-matched-element-of-a-list.aspx). Hence, the position of maximum element is: 3

# Python | Program to input, append and print the list elements

Here, we are going to learn how to **declare a list, input elements, append the elements in the list**and finally, print the list?  
Submitted by **IncludeHelp**, on July 26, 2018

**Read the value of N (limit of the list), input N elements and print the elements in Python.**

**Example:**

Input:

Enter limit of the list: 5

Enter an integer: 10

Enter an integer: 20

Enter an integer: 30

Enter an integer: 40

Enter an integer: 50

Output:

Input list elements are:

10

20

30

40

50

**Program:**

# declare a list

list = []

# read limit (value of n)

# for maximum number of elements

n = int (input ("Enter limit of the list: "))

# input n integer element

# and append to the list

**for** i **in** range (n) :

item = int (input ("Enter an integer: "))

list.append (item)

# print all elements

**print** "Input list elements are: "

**for** i **in** range (n) :

**print** list [i]

**Output**

Enter limit of the list: 5

Enter an integer: 10

Enter an integer: 20

Enter an integer: 30

Enter an integer: 40

Enter an integer: 50

Input list elements are:

10

20

30

40

50

# Python | Program to input, append and print the list elements

Here, we are going to learn how to **declare a list, input elements, append the elements in the list**and finally, print the list?  
Submitted by **IncludeHelp**, on July 26, 2018

**Read the value of N (limit of the list), input N elements and print the elements in Python.**

**Example:**

Input:

Enter limit of the list: 5

Enter an integer: 10

Enter an integer: 20

Enter an integer: 30

Enter an integer: 40

Enter an integer: 50

Output:

Input list elements are:

10

20

30

40

50

**Program:**

# declare a list

list = []

# read limit (value of n)

# for maximum number of elements

n = int (input ("Enter limit of the list: "))

# input n integer element

# and append to the list

**for** i **in** range (n) :

item = int (input ("Enter an integer: "))

list.append (item)

# print all elements

**print** "Input list elements are: "

**for** i **in** range (n) :

**print** list [i]

**Output**

Enter limit of the list: 5

Enter an integer: 10

Enter an integer: 20

Enter an integer: 30

Enter an integer: 40

Enter an integer: 50

Input list elements are:

10

20

30

40

50

# Python | Program to remove duplicate elements from the list

Here, we are going to learn how to **remove duplicate elements from the list in python**? To remove duplicate elements, we will append the unique elements to another list.  
Submitted by **IncludeHelp**, on July 26, 2018

**Example:**

Input:

list1: [10, 20, 10, 20, 30, 40, 30, 50]

Output:

List after removing duplicate elements

list2: [10, 20, 30, 40, 50]

**Logic:**

To implement the program is too easy, we have to append elements one by one to another list by checking whether element is available in the new list or not.

Let suppose, 20 is available three times in the list list1 and when we append 20 (first occurrence) to the list list2, it will be appended, but when we append 20 (second occurrence) to the list list2, condition will be false and item will not be appended. And finally, we will get list without duplicate elements.

**Program:**

# declare list

list1 = [10, 20, 10, 20, 30, 40, 30, 50]

# creating another list with unique elements

# declare another list

list2 = []

# appending elements

**for** n **in** list1:

**if** n **not** **in** list2:

list2.append(n)

# printing the lists

**print** "Original list"

**print** "list1: ", list1

**print** "List after removing duplicate elements"

**print** "list2: ", list2

**Output**

Original list

list1: [10, 20, 10, 20, 30, 40, 30, 50]

List after removing duplicate elements

list2: [10, 20, 30, 40, 50]

**Program (Defining User defines function):**

# Function to remove duplicates

**def** removeDuplicates (list1):

# declare another list

list2 = []

# appending elements

**for** n **in** list1:

**if** n **not** **in** list2:

list2.append (n)

**return** list2

# Main code

# declare a list

list1 = [10, 20, 10, 20, 30, 40, 30, 50]

# print the list

**print** "Original list: ", list1

**print** "List after duplicate remove: ", removeDuplicates (list1)

**Output**

Original list: [10, 20, 10, 20, 30, 40, 30, 50]

List after duplicate remove: [10, 20, 30, 40, 50]

# Python | Program to Create two lists with EVEN numbers and ODD numbers from a list

Here, we will learn how to **create two lists with EVEN and ODD numbers from a given list in Python**? To implement this program, we will check EVEN and ODD numbers and appends two them separate lists.  
Submitted by **IncludeHelp**, on July 26, 2018

**Given a list, and we have to create two lists 1) list with EVEN numbers and 2) list with ODD numbers from given list in Python.**

**Example:**

Input:

List1 = [11, 22, 33, 44, 55]

Output:

List with EVEN numbers: [22, 44]

List with ODD NUMBERS: [11, 33, 55]

**Logic:**

To create lists with EVEN and ODD numbers, we will traverse each element of list1 and append EVEN and ODD numbers in two lists by checking the conditions for EVEN and ODD.

**Program:**

# declare and assign list1

list1 = [11, 22, 33, 44, 55]

# declare listOdd - to store odd numbers

# declare listEven - to store even numbers

listOdd = []

listEven = []

# check and append odd numbers in listOdd

# and even numbers in listEven

**for** num **in** list1:

**if** num%2 == 0:

listEven.append(num)

**else**:

listOdd.append(num)

# print lists

**print** "list1: ", list1

**print** "listEven: ", listEven

**print** "listOdd: ", listOdd

**Output**

list1: [11, 22, 33, 44, 55]

listEven: [22, 44]

listOdd: [11, 33, 55]

# Python | Program to print all numbers which are divisible by M and N in the List

Here, we will learn how to **print all numbers from the string which are divisible by M an N in Python**?  
Submitted by **IncludeHelp**, on July 27, 2018

**Given a list of the integers and *M*, *N* and we have to print the numbers which are divisible by *M*, *N* in Python.**

**Example:**

Input:

List = [10, 15, 20, 25, 30]

M = 3, N=5

Output:

15

30

To find and print the list of the numbers which are divisible by M and N, we have to traverse all elements using a loop, and check whether the number is divisible by M and N, if number is divisible by M and N both print the number.

**Program:**

# declare a list of integers

list = [10, 15, 20, 25, 30]

# declare and assign M and N

M = 3

N = 5

# print the list

**print** "List is: ", list

# Traverse each element and check

# whether it is divisible by M, N

# or not, if condition is true print

# the element

**print** "Numbers divisible by {0} and {1}".format (M, N)

**for** num **in** list:

**if**( num%M==0 **and** num%N==0 ) :

**print** num

**Output**

List is: [10, 15, 20, 25, 30]

Numbers divisible by 3 and 5

15

# Python | Create three lists of numbers, their squares and cubes

Here, we are going to learn how to **create three lists with the numbers, their squares and cubes in Python, where range of the numbers is given**.  
Submitted by **IncludeHelp**, on August 03, 2018

**Take a range i.e. start and end, and we have to create three lists, list1 should contains numbers, list2 should contain squares of the numbers and list3 should contain cubes of the numbers in Python.**

**Example:**

Input:

Start = 1

End = 10

Output:

numbers: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

squares: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

cubes : [1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]

**Logic:**

* Declare three lists.
* Define range, here we are defining start with 1 and end with 10.
* Run a loop with the range as range(start, end+1) and loop counter as count.
* Append the loop counter count to the list named numbers, append square to the list named squares and append the cube to the list named cubes.
* Finally, print the lists.

**Program:**

# declare lists

numbers = []

squares = []

cubes = []

# start and end numbers

start = 1

end = 10

# run a loop from start to end+1

**for** count **in** range (start, end+1) :

numbers.append (count)

squares.append (count\*\*2)

cubes.append (count\*\*3)

# print the lists

**print** "numbers: ",numbers

**print** "squares: ",squares

**print** "cubes : ",cubes

**Output**

numbers: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

squares: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

cubes : [1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]

### **By defining own functions**

# define function to add numbers in list

**def** listNumbers(a,b):

#define dynamic list

list = []

**for** count **in** range(a,b+1):

list.append(count)

#return list

**return** list

# define function to add squares in list

**def** listSquares(a,b):

#define dynamic list

list = []

**for** count **in** range(a,b+1):

list.append(count\*\*2)

#return list

**return** list

# define function to add cubes in list

**def** listCubes(a,b):

#define dynamic list

list = []

**for** count **in** range(a,b+1):

list.append(count\*\*3)

#return list

**return** list

# Main code

# declare lists

numbers = []

squares = []

cubes = []

# start and end numbers

start = 1

end = 10

# get values in lists

numbers = listNumbers(start, end)

squares = listSquares(start, end)

cubes = listCubes(start, end)

# print the lists

**print** "numbers: ",numbers

**print** "squares: ",squares

**print** "cubes : ",cubes

**Output**

numbers: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

squares: [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]

cubes : [1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]

# Python | Create two lists with first half and second half elements of a list

Here, we are going to learn how to **create two lists with first half and second half elements of a given list in Python**?  
Submitted by **IncludeHelp**, on August 03, 2018

**Given a list, and we have to create two lists from first half elements and second half elements of a list in Python.**

**Example:**

Input:

list: [10, 20, 30, 40, 50, 60]

Output:

list1: [10, 20, 30]

list2: [40, 50, 60]

**Logic:**

* First take list (Here, we are taking list with 6 elements).
* To get the elements from/till specified index, use list[n1:n2] notation.
* To get first half elements, we are using list[:3], it will return first 3 elements of the list.
* And, to get second half elements, we are using list[3:], it will return elements after first 3 elements. In this example, we have only 6 elements, so next 3 elements will be returned.
* Finally, print the lists.

**Program:**

# define a list

list = [10, 20, 30, 40, 50, 60]

# Create list1 with half elements (first 3 elements)

list1 = list [:3]

# Create list2 with next half elements (next 3 elements)

list2 = list [3:]

# print list (s)

**print** "list : ",list

**print** "list1: ",list1

**print** "list2: ",list2

**Output**

list : [10, 20, 30, 40, 50, 60]

list1: [10, 20, 30]

list2: [40, 50, 60]

### **Using list[0:3] and list[3:6] instaed of list[:3] and list[3:]**

We can also use list[0:3] instead of list[:3] to get first 3 elements and list[3:6] instead of list[3:] to get next 3 elements after first 3 elements.

**Consider the program:**

# define a list

list = [10, 20, 30, 40, 50, 60]

# Create list1 with half elements (first 3 elements)

list1 = list [0:3]

# Create list2 with next half elements (next 3 elements)

list2 = list [3:6]

# print list (s)

**print** "list : ",list

**print** "list1: ",list1

**print** "list2: ",list2

**Output**

list : [10, 20, 30, 40, 50, 60]

list1: [10, 20, 30]

list2: [40, 50, 60]

### **By considering length of the list**

Let suppose list has n elements, then we can use list[0:n/2] and list[n/2:n].

**Consider the program:**

**If there are ODD numbers of elements in the list, program will display message "List has ODD number of elements." And exit.**

# define a list

list = [10, 20, 30, 40, 50, 60]

# get the length of the list

n = len(list)

# condition to check length is EVEN or not

# if lenght is ODD, show message and exit

**if**( n%2 != 0 ):

**print** "List has ODD number of elements."

exit()

# Create list1 with half elements (first 3 elements)

list1 = list [0:n/2]

# Create list2 with next half elements (next 3 elements)

list2 = list [n/2:n]

# print list (s)

**print** "list : ",list

**print** "list1: ",list1

**print** "list2: ",list2

**Output**

list : [10, 20, 30, 40, 50, 60]

list1: [10, 20, 30]

list2: [40, 50, 60]

Python | print list after removing EVEN numbers

Here, we are going to implement a **python program that will print the list after removing EVEN numbers**.  
Submitted by **IncludeHelp**, on August 06, 2018

**Given a list, and we have to print the list after removing the EVEN numbers in Python.**

**Example:**

Input:

list = [11, 22, 33, 44, 55]

Output:

list after removing EVEN numbers

list = [11, 33, 55]

**Logic:**

* Traverse each number in the list by using [for...in](https://www.includehelp.com/python/looping-constructs.aspx) loop.
* Check the condition i.e. checks number is divisible by 2 or not – to check EVEN, number must be divisible by 2.
* If number is divisible by 2 i.e. EVEN number, remove the number from the list.
* To remove the number from the list, use [list.remove() method](https://www.includehelp.com/python/remove-first-occurrence-of-a-given-element-in-the-list.aspx).

**Program:**

# list with EVEN and ODD number

list = [11, 22, 33, 44, 55]

# print original list

**print** "Original list:"

**print** list

# loop to traverse each element in the list

# and, remove elements

# which are EVEN (divisible by 2)

**for** i **in** list:

**if**(i%2 == 0):

list.remove(i)

# print list after removing EVEN elements

**print** "list after removing EVEN numbers:"

**print** list

Python | print list after removing ODD numbers

Here, we are going to implement a **python program that will print the list after removing ODD numbers**.  
Submitted by **IncludeHelp**, on August 06, 2018

**Given a list, and we have to print the list after removing the ODD numbers in Python.**

**Example:**

Input:

list = [11, 22, 33, 44, 55]

Output:

list after removing ODD numbers

list = [22, 44]

**Logic:**

* Traverse each number in the list by using [for...in](https://www.includehelp.com/python/looping-constructs.aspx) loop.
* Check the condition i.e. checks number is divisible by 2 or not – to check ODD, number must not be divisible by 2.
* If number is not divisible by 2 i.e. ODD number from the list, use [list.remove() method](https://www.includehelp.com/python/remove-first-occurrence-of-a-given-element-in-the-list.aspx).

**Program:**

# list with EVEN and ODD number

list = [11, 22, 33, 44, 55]

# print original list

**print** "Original list:"

**print** list

# loop to traverse each element in the list

# and, remove elements

# which are ODD (not divisible by 2)

**for** i **in** list:

**if**(i%2 != 0):

list.remove(i)

# print list after removing ODD elements

**print** "list after removing ODD numbers:"

**print** list

**Output**

Original list:

[11, 22, 33, 44, 55]

list after removing ODD numbers:

[22, 44]

Python | Input comma separated elements, convert into list and print

Here, we are going to learn how to **convert comma separated elements into list using Python**? Here, we will **input comma separate elements and convert into a list of integers**.  
Submitted by **IncludeHelp**, on August 08, 2018

**Input comma separated elements (integers), and converts it into list in Python.**

**Example:**

Input:

Enter comma separated integers: 10,20,30,40,50

Output:

list: ['10', '20', '30', '40', '50']

List (after converted each element to int)

list (li) : [10, 20, 30, 40, 50]

**Logic:**

* Input a comma - separated string using raw\_input() method.
* Split the elements delimited by comma (,) and assign it to the list, to split string, use string.split() method.
* The converted list will contains string elements.
* Convert elements to exact integers:
  + Traverse each number in the list by using [for...in](https://www.includehelp.com/python/looping-constructs.aspx) loop.
  + Convert number (which is in string format) to the integer by using int() method.
* Print the list.

**Program:**

# input comma separated elements as string

str = str (raw\_input ("Enter comma separated integers: "))

**print** "Input string: ", str

# conver to the list

list = str.split (",")

**print** "list: ", list

# convert each element as integers

li = []

**for** i **in** list:

li.append(int(i))

# print list as integers

**print** "list (li) : ", li

**Output**

Enter comma separated integers: 10,20,30,40,50

Input string: 10,20,30,40,50

list: ['10', '20', '30', '40', '50']

list (li) : [10, 20, 30, 40, 50]

# Python | Convert a string to integers list

Here, we will learn **how to convert a string (that contains digits only) to the integers list in Python**?  
Submitted by **IncludeHelp**, on September 19, 2018

**Given a string with digits and we have to convert the string to its equivalent list of the integers in Python.**

**Example:**

Input:

str1 = "12345"

Output:

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

Input:

str1 = "12345ABCD"

Output:

ValueError

**Note:** The string must contain only digits between 0 to 9, if there is any character except digits, the program will through a **ValueError**.

## **How to convert character (only digit) to integer?**

To convert a character (that is digit only like: **'0', '1', '2', '3', '4', '5', '6', '7', '8', '9'**) to integer, we use int() function - which is a library function in Python. int() returns integer value equivalent to given character i.e. digit in character form.

**print** (int('0'))

**print** (int('1'))

**print** (int('2'))

**print** (int('3'))

**print** (int('4'))

**print** (int('5'))

**print** (int('6'))

**print** (int('7'))

**print** (int('8'))

**print** (int('9'))

**Output**

0

1

2

3

4

5

6

7

8

9

### **Python program to convert a string to integers list**

Here, we have a string **"12345"** and we are converting it to **integers list [1, 2, 3, 4, 5]**.

# program to convert string to integer list

# language: python3

# declare a list

str1 = "12345"

# list variable to integeres

int\_list =[]

# converting characters to integers

**for** ch **in** str1:

int\_list.append(int(ch))

# printing the str\_list and int\_list

**print** ("str1: ", str1)

**print** ("int\_list: ", int\_list)

**Output**

str1: 12345

int\_list: [1, 2, 3, 4, 5]

### **ValueError**

If there is any character except the digits, there will be an error **"ValueError: invalid literal for int() with base 10"**.

**Program with Error:**

# program to convert string to integer list

# language: python3

# declare a list

str1 = "12345ABCD"

# list variable to integeres

int\_list =[]

# converting characters to integers

**for** ch **in** str1:

int\_list.append(int(ch))

# printing the str\_list and int\_list

**print** ("str1: ", str1)

**print** ("int\_list: ", int\_list)

**Output**

Traceback (most recent call last):

File "/home/main.py", line 12, in

int\_list.append(int(ch))

ValueError: invalid literal for int() with base 10: 'A'

Python program to find N largest and smallest elements from the list

**Python heapq.nlargest() and heapq.nsmallest() functions Examples**: Here, we are going to learn **how to find N largest and smallest elements?**  
Submitted by [Yash Khandelwal](https://www.includehelp.com/Members/Yash-Khandelwal.aspx), on April 13, 2019

Here, we learn how to find out the N largest and smallest elements from the list? Where, list and Ngiven by the user, N may be any value but less than the list length.

Description:

There are two ways,

**1. By defining a function:**

Procedure:

1. Define the function name largest\_ele and smallest\_ele.
2. Pass two arguments in a function (l,n) : l is the list and n is the number of elements.
3. Run the for loop n times
4. In the loop find the maximum of the given list and append it to another list
5. And after appending to another list remove the maximum element from the list

**By the inbuilt module heapq module**

If you are looking for the N smallest or largest items and N is small compared to the overall size of the collection, these functions provide superior performance.

1. Import the heapq module
2. Give the list
3. Now use the function heapq.nlargest(n,l) and heapq.nsmallest(n,l) from the module to find the largest and the smallest numbers.

**Python code:**

# N largest and smallest element in a list

# by function and by the help of heapq module

#function to find n largest element

**def** largest\_ele(l,n):

s=[]

**for** i **in** range(n):

s.append(max(l)) #append max of list in a new list

l.remove(max(l)) #remove max of list from the list

**print**('by largest\_ele function: ',s)

#function to find n largest element

**def** smallest\_ele(m,n):

t=[]

**for** i **in** range(n):

t.append(min(m))#append min of list in a new list

m.remove(min(m))#remove min of list from the list

**print**('by smallest\_ele function: ',t)

l=[2,4,6,8,10]

m=[0,1,2,3,4,5,6]

n=2

largest\_ele(l,n)

smallest\_ele(m,n)

# using the inbuilt module function

# heapq.nlargest and heapq.nsmallest

**import** heapq

nums = [1, 8, 2, 23, 7, -4, 18, 23, 42, 37, 2]

**print**('BY heapq.nlargest: ',heapq.nlargest(3, nums)) # Prints [42, 37, 23]

**print**('BY heapq.nsmallest: ',heapq.nsmallest(3, nums)) # Prints [-4, 1, 2]

**Output**

by largest\_ele function: [10, 8]

by smallest\_ele function: [0, 1]

BY heapq.nlargest: [42, 37, 23]

BY heapq.nsmallest: [-4, 1, 2]

**Note:** The **nlargest() and nsmallest() functions** are most appropriate if you are trying to find a relatively small number of items. If you are simply trying to find the single smallest or largest item (N=1), it is faster to use [min()](https://www.includehelp.com/python/find-minimum-value-from-given-parameters-using-min.aspx) and [max()](https://www.includehelp.com/python/find-maximum-value-from-given-parameters-using-max.aspx).

a = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]

for i in range(len(a)):

for j in range(len(a[i])):

print(a[i][j], end=' ')

print()

………………………

a = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]

for row in a:

for elem in row:

print(elem, end=' ')

print()

……………..

a = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]

s = 0

for i in range(len(a)):

for j in range(len(a[i])):

s += a[i][j]

print(s)

……………………….

a = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]

s = 0

for row in a:

for elem in row:

s += elem

print(s)

……………………..

**Accessing a multidimensional list:**

**Approach 1:**

filter\_none

edit

play\_arrow

brightness\_4

|  |
| --- |
| # Python program to demonstrate printing  # of complete multidimensional list  a = [[2, 4, 6, 8, 10], [3, 6, 9, 12, 15], [4, 8, 12, 16, 20]]  print(a) |

**Output:**

[[2, 4, 6, 8, 10], [3, 6, 9, 12, 15], [4, 8, 12, 16, 20]]

**Approach 2:** Accessing with the help of loop.

filter\_none

edit

play\_arrow

brightness\_4

|  |
| --- |
| # Python program to demonstrate printing  # of complete multidimensional list row  # by row.  a = [[2, 4, 6, 8, 10], [3, 6, 9, 12, 15], [4, 8, 12, 16, 20]]  for record in a:      print(record) |

**Output:**

[2, 4, 6, 8, 10]

[3, 6, 9, 12, 15]

[4, 8, 12, 16, 20]

**Approach 3:** Accessing using square brackets.  
Example:

filter\_none

edit

play\_arrow

brightness\_4

|  |
| --- |
| # Python program to demonstrate that we  # can access multidimensional list using  # square brackets  a = [ [2, 4, 6, 8 ],      [ 1, 3, 5, 7 ],      [ 8, 6, 4, 2 ],      [ 7, 5, 3, 1 ] ]    for i in range(len(a)) :      for j in range(len(a[i])) :          print(a[i][j], end=" ")      print() |

**Output:**

2 4 6 8

1 3 5 7

8 6 4 2

7 5 3 1

**Creating a multidimensional list with all zeros:**

filter\_none

edit

play\_arrow

brightness\_4

|  |
| --- |
| # Python program to create a m x n matrix  # with all 0s  m = 4  n = 5    a = [[0 for x in range(n)] for x in range(m)]  print(a) |

**Output:**

[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]

# Python Program to Add two Matrices

BY CHAITANYA SINGH | FILED UNDER: [PYTHON EXAMPLES](https://beginnersbook.com/category/python-examples/)

In this article, we will see **how to add two matrices in Python**. Before we see how to implement matrix addition in Python, lets see what it looks like:

M1 = [[1,1,1],

[1,1,1],

[1,1,1]]

M2 = [[1,2,3],

[4,5,6],

[7,8,9]]

Sum of these matrices:

= [[2,3,4],

[5,6,7],

[8,9,10]]

## Program for adding two matrices

To represent a matrix, we are using the concept of [nested lists](https://beginnersbook.com/2018/02/python-list/). All the elements of both the input matrices are represented as nested lists. All the elements of output list are initialized as zero.

We are iterating the matrix and adding the corresponding elements of both the given matrices and assigning the value in the output matrix.

# This program is to add two given matrices

# We are using the concept of nested lists to represent matrix

# first matrix

M1 = [[1, 1, 1],

[1, 1, 1],

[1, 1, 1]]

# second matrix

M2 = [[1, 2, 3],

[4, 5, 6],

[7, 8, 9]]

# In this matrix we will store the sum of above matrices

# we have initialized all the elements of this matrix as zero

sum = [[0, 0, 0],

[0, 0, 0],

[0, 0, 0]]

# iterating the matrix

# rows: number of nested lists in the main list

# columns: number of elements in the nested lists

for i in range(len(M1)):

for j in range(len(M1[0])):

sum[i][j] = M1[i][j] + M2[i][j]

# displaying the output matrix

for num in sum:

print(num)

………………………

m=int(input("enter rows"));

n=int(input("enter columns"));

#in python initilization is needed before indexing.

matrix1=[[0 for j in range(0,n)] for i in range(0,m)]   # matrix 1 initialization with 0s

matrix2=[[0 for j in range(0,n)] for i in range(0,m)]    #matrix 2 intialization with 0s

res\_matrix=[[0 for j in range(0,n)] for i in range(0,m)] # matrix for storing result

print("enter first matrix elements")

for i in range(0,m):

for j in range(0,n):

matrix1[i][j]= int(input("enter an element"))

print("enter second matrix elements ")

for i in range(0,m):

for j in range(0,n):

matrix2[i][j]=int(input("enter an element"))

for i in range(0,m):

for j in range(0,n):

res\_matrix[i][j]=matrix1[i][j]+matrix2[i][j]

……………….

1. X = [[1,2,3],
2. [4,5,6],
3. [7,8,9]]
5. Y = [[10,11,12],
6. [13,14,15],
7. [16,17,18]]
9. Result = [[0,0,0],
10. [0,0,0],
11. [0,0,0]]
13. # iterate through rows of X
14. **for** i **in** range(len(X)):
15. **for** j **in** range(len(Y[0])):
16. **for** k **in** range(len(Y)):
17. result[i][j] += X[i][k] \* Y[k][j]
18. **for** r **in** result:
19. **print**(r)

……………………

1. X = [[1,2],
2. [4,5],
3. [7,8]]
5. Result = [[0,0,0],
6. [0,0,0]]
8. # iterate through rows
9. **for** i **in** range(len(X)):
10. **for** j **in** range(len(X[0])):
11. result[j][i] = X[i][j]
13. **for** r **in** res
14. ult:
15. **print**(r)

………………………

Given two user input matrix. Our task is to display the addition of two matrix. In these problem we use nested List comprehensive.

## Algorithm

Step1: input two matrix.

Step 2: nested for loops only to iterate through each row and columns.

Step 3: At each iterationshall add the corresponding elements from two matrices and shall store the result.

## Example code

# Program to add two matrices using nested loop

A=[]

n=int(input("Enter N for N x N matrix : ")) #3 here

#use list for storing 2D array

#get the user input and store it in list (here IN : 1 to 9)

print("Enter the element ::>")

for i in range(n):

row=[] #temporary list to store the row

for j in range(n):

row.append(int(input())) #add the input to row list

A.append(row) #add the row to the list

print(A)

# [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

#Display the 2D array

print("Display Array In Matrix Form")

for i in range(n):

for j in range(n):

print(A[i][j], end=" ") #new line

print()

B=[]

n=int(input("Enter N for N x N matrix : ")) #3 here

#use list for storing 2D array

#get the user input and store it in list (here IN : 1 to 9)

print("Enter the element ::>")

for i in range(n):

row=[] #temporary list to store the row

for j in range(n):

row.append(int(input())) #add the input to row list

B.append(row) #add the row to the list

print(B)

# [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

#Display the 2D array

print("Display Array In Matrix Form")

for i in range(n):

for j in range(n):

print(B[i][j], end=" ")

print() #new line

result = [[0,0,0], [0,0,0], [0,0,0]]

# iterate through rows

for i in range(n):

# iterate through columns

for j in range(len(A[0])):

result[i][j] = A[i][j] + B[i][j]

print("Resultant Matrix is ::>")

for r in result:

print("Resultant Matrix is ::>",r)

……………………