**NAME– TAUSIF KHAN**

**ROLL NO- 12**

**SUBJECT- OSTL**

**S.E COMPS- B**

**Experiment No- 2**

**Aim**: Write a program to demonstrate use of list in python:

a) Put Even and Odd elements in two different lists.

b) Merge and sort the two list.

c) Update first element with X value and delete the middle element of list.

d) Find max and min element from the list.

e) Add N names into the existing number list and check if word python is present in list.

**Tools Used**: Python 3.4.3, Terminal

**Theory**:

1. Explain the basic of List and its syntax ?

Ans: List is a collection which is ordered and changeable. Allows duplicate members.

A list is a collection which is ordered and changeable. In Python lists are written with square brackets.

Eg: Create a list

thislist = ["apple", "banana", "cherry"]  
print(thislist)

1. Explain creating list using range function ?

Ans: Often times we want to create a list containing a continuous value like, in a range of 100-200. But the result is not exactly what we were expecting because Python does not unpack the result of the range() function. We can use argument-unpacking operator i.e. \*.

Eg: # Create a list in a range of 10-20

Code:

My\_list = [\*range(10, 21, 1)]

# Print the list

print(My\_list)

Output:

[10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

1. How to update list and concatenate list ?

Ans: Update a list: update() function in set adds elements from a set (passed as an argument) to the set.

Syntax :  
set1.update(set2)  
Here set1 is the set in which set2 will be added.

Parameters :  
Update() method takes only a single argument. The single argument can be a set, list, tuples or a dictionary. It automatically converts into a set and adds to the set.

Concatenate list: We traverse the second list and keep appending elements in the first list, so that first list would have all the elements in both lists and hence would perform the append.

Eg:

test\_list1 = [1, 4, 5, 6, 5]

test\_list2 = [3, 5, 7, 2, 5]

for i in test\_list2 :

test\_list1.append(i)

print ("Concatenated list using naive method : "

+ str(test\_list1))

Output:

Concatenated list using naive method : [1, 4, 5, 6, 5, 3, 5, 7, 2, 5]

1. Explain aliasing of list with syntax ?

Ans: In computing, aliasing describes a situation in which a data location in memory can be accessed through different symbolic names in the program. Thus, modifying the data through one name implicitly modifies the values associated with all aliased names, which may not be expected by the programmer. As a result, aliasing makes it particularly difficult to understand, analyze and optimize programs. Aliasing analysers intend to make and compute useful information for understanding aliasing in programs

Aliasing. Because the same list has two different names, a and b , we say that it is aliased. Changes made with one alias affect the other. In the codelens example below, you can see that a and b refer to the same list after executing the assignment statement b = a

Syntax:

alias [-p] [name[=value] ... ]

1. Explain cloning of list with syntax ?

Ans: This method is considered when we want to modify a list and also keep a copy of the original. In this we make a copy of the list itself, along with the reference. This process is also called cloning.

Eg:

# Python program to copy or clone a list

# Using the Slice Operator

def Cloning(li1):

li\_copy = li1[:]

return li\_copy

# Driver Code

li1 = [4, 8, 2, 10, 15, 18]

li2 = Cloning(li1)

print("Original List:", li1)

print("After Cloning:", li2)

Output:

Output:  
Original List: [4, 8, 2, 10, 15, 18]  
After Cloning: [4, 8, 2, 10, 15, 18]

1. List methods an explain ?

Ans:

|  |  |
| --- | --- |
| Method | Description |
| append() | Adds element at end of the List. |
| clear() | Removes all elements from the list. |
| copy() | Returns a copy of the List. |
| count() | Returns the number of elements with the specified value |
| extend() | Add the elements of a list (or any iterable), to the end of the current list |
| index() | Returns the index of the first element with the specified value |
| insert() | Adds an element at the specified position |
| pop() | Removes the element at the specified position |
| remove() | Removes the first item with the specified value |
| reverse() | Reverses the order of the list |
| sort() | Sorts The list. |

1. Explain nested list with example ?

Ans: A list can contain any sort object, even another list (sublist), which in turn can contain sublists themselves, and so on. This is known as nested list. You can use them to arrange data into hierarchical structures.

Eg:

matrix = [[j for j in range(5)] for i in range(5)]

print(matrix)

Output:

[[0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4], [0, 1, 2, 3, 4]

**CODE & OUTPUT:-**

**Put Even and Odd elements in two different lists.**

**Code:**

a=[]

n=int(input("Enter number of elements:"))

for i in range(1,n+1):

b=int(input("Enter element:"))

a.append(b)

even=[]

odd=[]

for j in a:

if(j%2==0):

even.append(j)

else:

odd.append(j)

print("The even list",even)

print("The odd list",odd)

**Output:**

Enter number of elements: 5Enter element: 23Enter element: 32Enter element: 40Enter element: 65Enter element: 77The even list [32, 40] The odd list [23, 65, 77]

**Merge and sort the two list.**

**Code:**

a=[]

c=[]

n1=int(input("Enter number of elements:"))

for i in range(1,n1+1):

b=int(input("Enter element:"))

a.append(b)

n2=int(input("Enter number of elements:"))

for i in range(1,n2+1):

d=int(input("Enter element:"))

c.append(d)

new=a+c

new.sort()

print("Sorted list is:",new)

**Output:**

Enter number of elements: 2Enter element: 4Enter element: 6Enter number of elements: 3Enter element: 4Enter element: 6Enter element: 3Sorted list is: [3, 4, 4, 6, 6]

**Find max and min element from the list.**

**Code:**

lst = []

num = int(input('How many numbers: '))

for n in range(num):

numbers = int(input('Enter number '))

lst.append(numbers)

print("Maximum element in the list is :", max(lst), "\nMinimum element in the list is :", min(lst))

**Output:**

How many numbers: 4Enter number 32Enter number 45Enter number 7Enter number 65Maximum element in the list is : 65 Minimum element in the list is : 7

**Update first element with X value and delete the middle element of list.**

**Code:**

lst = [3, 5, 7]

lst.insert(0,9)

lst.remove(5)

lst

Output:

[9, 3, 7]

**Add N names into the existing number list and check if word python is present in list.**

**Code:**

lst = ["OS", "AOA" , "CG" ,"Maths"]

lst.insert(0,"Python")

print(lst)

for i in lst:

if (i=="Python"):

print("Python is present")

**Output:**

['Python', 'OS', 'AOA', 'CG', 'Maths'] Python is present

**CONCLUSION** – Thus we have learned how to implement list in python.