- extra program

In [3]:
```
list1 = []
n = 0
while n != 4:
    print("1. for add a value")
    print("2. for delete a value")
    print("3. for display list")
    print("4. exit")
    n = int(input("Enter choice : "))

    if n == 1:
        l1 = int(input("Enter value that you want to add :"))
        list1.append(l1)
        print("list is : ",list1)

    elif n == 2:
        if len(list1)==0:
            print("add value first ")
        else:
            list1.pop()
            print("list is : ",list1)

    elif n == 3:
        print("list is : ",list1)

    else:
        print("final list is : ",list1)
        break
```

```
1. for add a value
2. for delete a value
3. for display list
4. exit
Enter choice : 1
Enter value that you want to add :10
list is :  [10]
1. for add a value
2. for delete a value
3. for display list
4. exit
Enter choice : 1
Enter value that you want to add :20
list is :  [10, 20]
1. for add a value
2. for delete a value
3. for display list
4. exit
Enter choice : 2
list is :  [10]
1. for add a value
2. for delete a value
3. for display list
4. exit
Enter choice : 3
list is :  [10]
1. for add a value
2. for delete a value
```

```
3. for display list
4. exit
Enter choice : 4
final list is :  [10]
```

module = 3

(1) What is List? How will you reverse a list?

- Lists are used to store multiple items in a single variable.
- Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.
- You can reverse a list in Python using the built-in reverse() or reversed() methods.
- These methods will reverse the list without creating a new list.
- Python reverse() and reversed() will reverse the elements in the original list object.
- Reversing a list is a common part of any programming language.

(2) How will you remove last object from a list?

- You can use list1.pop() method to remove the last element from the list.

(3) Suppose list1 is [2, 33, 222, 14, and 25], what is list1 [-1]?

- list[-1] = 25

(4) Differentiate between append () and extend () methods?

- append() :
- For appending to a list, append() function is used.
- The append() function can add a single element to the end of a list.
- After append(), the length of the list will increase by 1 element only.
- extend() :
- For extending a list, extend() function is used.
- The extend() function can add multiple elements from a list supplied to it as argument.
- After extend() the length of the list will increase by the length of the list given as argument to extend()

(5) Write a Python function to get the largest number, smallest num and sum of all from a list.

In [9]:
```
list1 = []
n = int(input("how many number you want to enter : "))
for i in range(n):
    num = int(input("Enter number : "))
    list1.append(num)
print("final list is :",list1)
print("maximum element of list is : ",max(list1))
print("minimum element of list is : ",min(list1))
sum = 0
for i in list1:
    sum = sum + i
print("sum of the all elements of list is : ",sum)
```

```
how many number you want to enter : 4
Enter number : 1
Enter number : 2
Enter number : 3
Enter number : 5
final list is : [1, 2, 3, 5]
maximum element of list is :  5
minimum element of list is :  1
sum of the all elements of list is :  11
```

(6) How will you compare two lists?

- The sort() method with == operator
- Python sort() function is used to sort the lists.
- The same list's elements are the same index position it means; lists are equal.

(7) Write a Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings.

In [13]:
```
list1 = ['a','ab','aab','bb']
count = 0
for i in list1:
    if len(list1)>=2 and i[0]==i[-1]:
        count += 1
print(count)
```

```
2
```

(8) Write a Python program to remove duplicates from a list.

In [32]:
```
ist1 = [5,1,2,3,2,2,3,4,4,4]
list2=[]
for i in list1:
    if list1.count(i)>1:
        list2.append(i)
        a=list1.count(i)
        for j in range(a):
            b=list1.index(i)
            list1.pop(b)
    else:
        list2.append(i)
print(list2)
```

```
[1, 2, 3, 4, 5]
```

(9) Write a Python program to check a list is empty or not.

In [20]:
```
list1 = ['kris','tax','dhruv','kris']
if len(list1) == 0:
    print('List is empty')
else:
    print('List not empty')
```

```
List not empty
```

(10) Write a Python function that takes two lists and returns true if they have at least one common member.

In [18]:
```
list1 = []
list2 = []
n = input("enter the element of the list : ").split()
k = input("enter the element of the list : ").split()
list1.extend(n)
list2.extend(k)
print(list1)
print(list2)
for i in list1:
    if i in list2:
        print("True")
        break
    else:
        pass
```

```
enter the element of the list : 1 2 3
enter the element of the list : 2 3 4
['1', '2', '3']
['2', '3', '4']
True
```

(11) Write a Python program to generate and print a list of first and last 5 elements where the values are square of numbers between 1 and 30.

In [39]:
```
list1 = []
for i in range(1,31):
    list1 += [i**2]
print(list1)
print(list1[:5])
print(list1[-5:])
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324,
361, 400, 441, 484, 529, 576, 625, 676, 729, 784, 841, 900]
[1, 4, 9, 16, 25]
[676, 729, 784, 841, 900]
```

(12) Write a Python function that takes a list and returns a new list with unique elements of the first list.

In [19]:
```
list1 = [1,2,3,4,5,6,7,7,7,5,6]
list2 = []
for i in list1:
    if list1.count(i) == 1:
        list2.append(i)
    else:
        pass
print("list2 is :",list2)
```

```
list2 is : [1, 2, 3, 4]
```

(13) Write a Python program to convert a list of characters into a string.

In [8]:
```
list1 = ['k', 'r', 'i', 's','h','n','a']
str1 = ''.join(list1)
print(str1)
```

```
krishna
```

(14) Write a Python program to select an item randomly from a list.

In [78]:
```
import random
random.choice([1,2,2,5,4,78,58,69])
```

Out[78]:
```
4
```

(15) Write a Python program to find the second smallest number in a list.

In [7]:
```
list1 = []
n = int(input("how many number you want to enter : "))
for i in range(n):
    num = int(input("Enter number : "))
    list1.append(num)
list1.sort()
print("final list is :",list1)
print("The second smallest value of this list: ",list1[1])
```

```
how many number you want to enter : 5
Enter number : 12
Enter number : 25
Enter number : 140
Enter number : 10
Enter number : 45
final list is : [10, 12, 25, 45, 140]
The second smallest value of this list:  12
```

(16) Write a Python program to get unique values from a list.

In [20]:
```
my_list = [10, 20, 30, 40, 20, 50, 60, 40]
print("Original List : ",my_list)
my_set = set(my_list)
print("List of unique numbers : ",my_set)
```

```
Original List :  [10, 20, 30, 40, 20, 50, 60, 40]
List of unique numbers :  {40, 10, 50, 20, 60, 30}
```

(17) Write a Python program to check whether a list contains a sub list.

In [77]:
```
list1=[1,2,4,5,[1,2]]
for i in list1:
    if type(i)==type(list1):
        print("true")
```

```
true
```

(18) Write a Python program to split a list into different variables.

In [35]:
```
list1 = []
n = input("Enter a list : ").split()
list1.extend(n)
print(list1)
for i,j in zip(range(0,len(list1)+1),list1):
    print(f"Value of {i} is")
    i=j
    print(i)
```

```
Enter a list : 45 78 96 85 94
['45', '78', '96', '85', '94']
Value of 0 is
45
Value of 1 is
78
Value of 2 is
96
Value of 3 is
85
Value of 4 is
94
```

(19) What is tuple? Difference between list and tuple.

- Tuples are used to store multiple items in a single variable.
- Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Set, and Dictionary, all with different qualities and usage.
- A tuple is a collection which is ordered and unchangeable.
- The key difference between the tuples and lists is that while the tuples are immutable objects the lists are mutable .
- This means that tuples cannot be changed while the lists can be modified.

(20) Write a Python program to create a tuple with different data types.

In [23]:
```
tup1 = (1,2,[4,5,6],"tops",True)
print(tup1)
```

```
(1, 2, [4, 5, 6], 'tops', True)
```

(21) Write a Python program to create a tuple with numbers.

In [84]:
```
tup1 = (1,2,3,4,5,6)
print(tup1)
```

```
(1, 2, 3, 4, 5, 6)
```

(22) Write a Python program to convert a tuple to a string.

In [93]:
```
tup1 = ('h','e','l','l','o')
str = ''.join(tup1)
print (str)
```

```
hello
```

(23) Write a Python program to check whether an element exists within a tuple.

In [4]:
```
tup1 = (1,2,3,"String",True)
print(1 in tup1)
print(5 in tup1)
```

```
True
False
```

(24) Write a Python program to find the length of a tuple.

In [97]:
```
tup1 = ('krishna', 'dhruv', 'taksh')
print("length of tuple is : ",len(tup1))
```

```
length of tuple is :  3
```

(25) Write a Python program to convert a list to a tuple.

In [2]:
```
list1 = [5, 10, 7, 4, 15, 3]
print(list1)
tup1 = tuple(list1)
print(tup1)
```

```
[5, 10, 7, 4, 15, 3]
(5, 10, 7, 4, 15, 3)
```

(26) Write a Python program to reverse a tuple.

In [3]:
```
tup1 = (2, 4, 6, 8, 10, 12)
x = reversed(tup1)
print(tuple(x))
```

```
(12, 10, 8, 6, 4, 2)
```

(27) Write a Python program to replace last value of tuples in a list.

In [5]:
```
l = [(10, 20, 40), (40, 50, 60), (70, 80, 90)]
print([t[:-1] + (100,) for t in l])
```

```
[(10, 20, 100), (40, 50, 100), (70, 80, 100)]
```

(28) Write a Python program to find the repeated items of a tuple.

In [10]:
```
list = [1,2,3,3,56,56,7,8,9,9]
new = []
for a in list:
    n = list.count(a)
    if n > 1:
        if new.count(a) == 0:
            new.append(a)
print(new)
```

```
[3, 56, 9]
```

(29) Write a Python program to remove an empty tuple(s) from a list of tuples.

In [12]:
```
list1 = [(),(1,2),(1,),()]
for i in list1:
    if len(i) == 0:
        list1.remove(i)
    else:
        pass
print(list1)
```

```
[(1, 2), (1,)]
```

(30) Write a Python program to unzip a list of tuples into individual lists.

In [3]:
```
list1 = [(),(1,2),(1,),()]
for i in list1:
    print(list(i))
```

```
[]
[1, 2]
[1]
[]
```

(31) Write a Python program to convert a list of tuples into a dictionary.

In [8]:
```
list_1=[("Nakul",93), ("Shivansh",45), ("Samved",65)]
dict_1=dict()
for i,j in list_1:
    dict_1.setdefault(i, []).append(j)
print(dict_1)
```

```
{'Nakul': [93], 'Shivansh': [45], 'Samved': [65]}
```

(32) How will you create a dictionary using tuples in python?

- To convert a tuple to dictionary in Python, use the dict() method.
- A dictionary object can be constructed using a dict() function.
- The dict() function takes a tuple of tuples as an argument and returns the dictionary.

(33) Write a Python script to sort (ascending and descending) a dictionary by value.

In [8]:
```
dict1 = {4: 'three',2: 'two',3: 'three'}
dict2 = sorted(dict1.items())
dict3 = sorted(dict1.items(),reverse=True)
print("Sorted dictionary in ascending form is :",dict2)
print("Sorted dictionary in descending form is :",dict3)
```

```
Sorted dictionary in ascending form is : [(2, 'two'), (3, 'three'), (4, 'thre
e')]
Sorted dictionary in descending form is : [(4, 'three'), (3, 'three'), (2, 'tw
o')]
```

(34) Write a Python script to concatenate following dictionaries to create a new one.

In [10]:
```
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
dic4 = {}
for d in (dic1, dic2, dic3): dic4.update(d)
print(dic4)
```

```
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

(35) Write a Python script to check if a given key already exists in a dictionary.

In [2]:
```
dict1 = {1:"one", 2:"two", 3:"three"}
key = 4
if key in dict1:
    print("key is exist.")
else:
    print("key does not exist.")
```

```
key does not exist.
```

(36) How Do You Traverse Through A Dictionary Object In Python?

- There are two ways of iterating through a Python dictionary object.
- One is to fetch associated value for each key in keys() list.
- There is also items() method of dictionary object which returns list of tuples, each tuple having key and value.

(37) How Do You Check The Presence Of A Key In A Dictionary.

- has given key already exists in a dictionary.
- has_key() method returns true if a given key is available in the dictionary, otherwise it returns a false.
- With the Inbuilt method has_key() , use if statement to check if the key is present in the dictionary or not.

(38) Write a Python script to print a dictionary where the keys are numbers between 1 and 15.

In [3]:
```
dict1 = dict()
for x in range(1,16):
    dict1[x]=x+2
print(dict1)
```

```
{1: 3, 2: 4, 3: 5, 4: 6, 5: 7, 6: 8, 7: 9, 8: 10, 9: 11, 10: 12, 11: 13, 12: 1
4, 13: 15, 14: 16, 15: 17}
```

(39) Write a Python program to check multiple keys exists in a dictionary.

In [4]:
```
dict1 = {1: 'one',6: 'six',3: 'three'}
if  len(dict1.keys())>1:
    print("The MUltiple keys is Present.")
else:
    print("The multiple key is not Present.")
```

```
The MUltiple keys is Present.
```

(40) Write a Python script to merge two Python dictionaries.

In [5]:
```
dict1 = {1: 'one',6: 'six',3: 'three'}
dict2 = {2: 'two',4: 'four',5: 'five'}
dict1.update(dict2)
print("The Merging of two Dict Is: ",dict1)
```

```
The Merging of two Dict Is:  {1: 'one', 6: 'six', 3: 'three', 2: 'two', 4: 'fou
r', 5: 'five'}
```

(41) Write a Python program to map two lists into a dictionary.

In [7]:
```
list1 = ['hi','hello','tops','tech']
list2 = [11,22,33,66]
dict1 = dict(zip(list1,list2))
print(dict1)
```

```
{'hi': 11, 'hello': 22, 'tops': 33, 'tech': 66}
```

(42) Write a Python program to combine two dictionary adding values for common keys.

- d1 = {'a': 100, 'b': 200, 'c':300}
- d2 = {'a': 300, 'b': 200,'d':400}

In [8]:
```
dict1 = {'a': 100, 'b': 200, 'c':300}
dict2 = {'a': 300, 'b': 200, 'd':400}
for key in dict2:
    if key in dict1:
        dict2[key] = dict2[key] + dict1[key]
dict1.update(dict2)
print(dict1)
```

```
{'a': 400, 'b': 400, 'c': 300, 'd': 400}
```

(43) Write a Python program to print all unique values in a dictionary.

In [11]:
```
dict1 = {'a':100, 'b':200, 'c':300, 'a':500}
dict2 =set(dict1)
print(dict1)
```

```
{'a': 500, 'b': 200, 'c': 300}
```

(44) Why Do You Use the Zip () Method in Python?

- Python's zip() function creates an iterator that will aggregate elements from two or more iterables.
- You can use the resulting iterator to quickly and consistently solve common programming problems, like creating dictionaries.

(45) Write a Python program to create and display all combinations of letters,selecting each letter from a different key in a dictionary.

- Sample data: {'1': ['a','b'], '2': ['c','d']}
- Expected Output:
- ac ad bc bd

In [14]:
```
dict1 = {'1': ['a','b'], '2': ['c','d']}
list1 = list(dict1.values())
for i in list1[0]:
    for j in list1[1]:
        print(i+j,end=" ")
```

```
ac ad bc bd
```

(46) Write a Python program to find the highest 3 values in a dictionary.

In [34]:
```
dict1 = eval(input("Enter a dictionary :-"))
val = list( dict1.values() )
val.sort()
print("Highest 3 values ",val[ - 1 : - 4 : - 1])
```

```
Enter a dictionary :-{ "Portal":16, "Express":14, "Path":15, "Walla":10,"Pytho
n":19}
Highest 3 values  [19, 16, 15]
```

(47) Write a Python program to combine values in python list of dictionaries.

- Sample data: [{'item': 'item1', 'amount': 400}, {'item': 'item2', 'amount': 300},{'item': 'item1', 'amount': 750}]
- Expected Output: Counter ({'item1': 1150, 'item2': 300})

In [40]:
```
a=[{'item': 'item1', 'amount': 400}, {'item': 'item2', 'amount': 300}, {'item':
dict1={}
val=0
for d in a:
    if d['item'] not in dict1:
        dict1[d['item']]=d['amount']
    else:
        dict1[d['item']]+=d['amount']
print(dict1)
```

```
{'item1': 1150, 'item2': 300}
```

(48) Write a Python program to create a dictionary from a string.

- Note: Track the count of the letters from the string. Sample string:'w3resource'
- Expected output: {'3': 1,'s': 1, 'r': 2, 'u': 1, 'w': 1, 'c': 1, 'e': 2, 'o': 1}

In [2]:
```
str1 = 'w3resource'
dict1 = {}
for letter in str1:
    dict1[letter] = dict1.get(letter, 0) + 1
print(dict1)
```

```
{'w': 1, '3': 1, 'r': 2, 'e': 2, 's': 1, 'o': 1, 'u': 1, 'c': 1}
```

(49) Write a Python function to calculate the factorial of a number (a non-negative integer).

In [1]:
```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
n=int(input("enter value find the fact : "))
print(factorial(n))
```

```
enter value find the fact : 3
6
```

(50) Write a Python function to check whether a number is in a given range.

In [7]:
```
def test_range(n):
    if n in range(3,9):
        print(f"{n} is in the range.")
    else :
        print("The number is outside the given range.")
test_range(5)
```

```
5 is in the range
```

(51) Write a Python function to check whether a number is perfect or not.

In [13]:
```
def perfect_number(n):
    sum = 0
    for x in range(1, n):
        if n % x == 0:
            sum += x
    return sum == n
print(perfect_number(6))
```

```
True
```

(52) Write a Python function that checks whether a passed string is palindrome or not.

In [10]:
```
def isPalindrome(string):
    left_pos = 0
    right_pos = len(string) - 1
    while right_pos >= left_pos:
        if not string[left_pos] == string[right_pos]:
            return False
        left_pos += 1
        right_pos -= 1
    return True
print(isPalindrome('geekeeg'))
```

```
True
```

(53) How do you perform pattern matching in Python? Explain

- Python Provides a module referred as re for performing pattern matching using regular

expression operations.
- Using re.search() Function
- Using re.match() Function
- Using re.fullmatch() Function
- Using re.findall() Function
- Using re.finditer() Function

(54) What is lambda function in python? What we call a function which is incomplete version of a function?

- A lambda function is a small anonymous function.
- A lambda function can take any number of arguments, but can only have one expression.
- Syntax : lambda arguments : expression
- Stub Function A function developed using bottom-up approach A function developed using top-down approach

(55) How Many Basic Types Of Functions Are Available In Python?

- 1). Built-in Functions:
- Built-in functions are the functions that are already written or defined in python.
- We only need to remember the names of built-in functions and the parameters used in the functions.
- As these functions are already defined so we do not need to define these functions. Below are some built-in functions of Python.
- 2). User-Defined Functions:
- The functions defined by a programmer to reduce the complexity of big problems and to use that function according to their need.
- This type of functions is called user-defined functions.

(56) How can you pick a random item from a list or tuple?

In [1]:
```
from random import choice
print("choice([1, 2, 3, 5, 9]) : ", choice([1, 2, 3, 5, 9]))
```

```
choice([1, 2, 3, 5, 9]) :  9
```

(57) How can you pick a random item from a range?

In [2]:
```
import random
x=random.randrange(10,20,2)
x
```

Out[2]:
```
18
```

(58) How can you get a random number in python?

- Python defines a set of functions that are used to generate or manipulate random numbers through the random module.
- choice() :- choice() is an inbuilt function in the Python programming language that returns a random item from a list, tuple, or string.
- randrange(beg, end, step):- The random module offers a function that can generate random numbers from a specified range and also allowing rooms for steps to be included, called randrange().
- random():- This method is used to generate a float random number less than 1 and greater or equal to 0.

(59) How will you set the starting value in generating random numbers?

- The random number generator needs a number to start with (a seed value), to be able to generate a random number.
- By default the random number generator uses the current system time.
- Use the seed() method to customize the start number of the random number generator.

(60) How will you randomizes the items of a list in place?

In [3]:
```
import random
list = [1,2,3,4,5,6,7,8,9];
random.shuffle(list)
print("Reshuffled list : ",list)
```

```
Reshuffled list :  [9, 7, 6, 2, 5, 3, 4, 8, 1]
```

(61) Write a Python program to read a random line from a file.

In [33]:
```
import random
def random_line(name):
    lines = open(name).read().splitlines()
    return random.choice(lines)
print(random_line('krishna.txt'))
```

```
name : taksh
```

(62) Write a Python program to convert degree to radian.

In [34]:
```
import math
deg = float(input("Enter Degree : "))
rad = deg*(math.pi/180)
print(f"{deg} in radian is : ",rad)
```

```
Enter Degree : 27
27.0 in radian is :  0.47123889803846897
```

(63) Write a Python program to calculate the area of a trapezoid.

In [36]:
```
a = float(input('Please Enter the First Base of a Trapezoid: '))
b = float(input('Please Enter the Second Base of a Trapezoid: '))
height = float(input('Please Enter the Height of a Trapezoid: '))
Area = 0.5 * (a + b) * height
print("Area of a Trapezium is = %.2f" %Area)
```

```
Please Enter the First Base of a Trapezoid: 5
Please Enter the Second Base of a Trapezoid: 8
Please Enter the Height of a Trapezoid: 6
Area of a Trapezium is = 39.00
```

(64) Write a Python program to calculate the area of a parallelogram.

In [37]:
```
parallelBase = float(input("Enter Parallelogram Base : "))
ParallelHeight = float(input("Enter Parallelogram Height : "))
parallelArea = parallelBase * ParallelHeight
print("The Area of a Parallelogram =", "%.3f" %parallelArea)
```

```
Enter Parallelogram Base : 20
Enter Parallelogram Height : 30
The Area of a Parallelogram = 600.000
```

(65) Write a Python program to calculate surface volume and area of a cylinder.

In [38]:
```
pi=22/7
height = float(input('Height of cylinder: '))
radian = float(input('Radius of cylinder: '))
volume = pi * radian * radian * height
sur_area = ((2*pi*radian) * height) + ((pi*radian**2)*2)
print("Volume is: %.3f" %volume)
print("Surface Area is: %.3f" %sur_area)
```

```
Height of cylinder: 25
Radius of cylinder: 15
Volume is: 17678.571
Surface Area is: 3771.429
```

(66) Write a Python program to returns sum of all divisors of a number.

In [39]:
```
def sum_div(n):
    divisors = [1]
    for i in range(2, n):
        if (n % i)==0:
            divisors.append(i)
    print(f"divisors of {n} is ",divisors)
    return sum(divisors)
n = int(input("Enter Number : "))
print(f"sum of divisors is :",sum_div(n))
```

```
Enter Number : 24
divisors of 24 is  [1, 2, 3, 4, 6, 8, 12]
sum of divisors is : 36
```

(67) Write a Python program to find the maximum and minimum numbers from the specified decimal numbers.

In [1]:
```
x = list(map(float, input("Enter multiple values: ").split()))
print(max(x))
print(min(x))
```

```
Enter multiple values: 10 20 30
30.0
10.0
```