```
>>> abs(-9)
9

>>> hash("Sudhamshu B N")
1300149873

>>> v = memoryview(b'abcefg')
>>> v[1]
98

>>> set(["a", "b", "c"])
{'a', 'c', 'b'}

>>> mylist = [1,2,2]
>>> all(mylist)
True

>>> dict(x=5, y=0)
{'x': 5, 'y': 0}

>>> help(abs)
Help on built-in function abs in module builtins:

abs(x, /)
    Return the absolute value of the argument.

>>> min([1,2,34,56,7,8,90,0])
0

>>> mylist = [1,2,3]
>>> any(mylist)
True

>>> dir()
['__annotations__',    '__builtins__',    '__doc__',    '__loader__',    '__name__',    '__package__',
'__spec__', 'a', 'foo']

>>> hex(367)
'0x16f'

>>> mylist = iter(["apple", "banana", "cherry"])
>>> next(mylist)
'apple'
>>> next(mylist)
```

```
'banana'
>>> next(mylist)
'Cherry'

>>> a = ("a", "b", "c", "d", "e", "f", "g", "h")
>>> a[slice(3)]
('a', 'b', 'c')

>>> ascii(['Pythön','Sudhamshu'])
"['Pyth\\xf6n', 'Sudhamshu']"

>>> divmod(21,14)
(1, 7)

>>> id("A")
55791168

>>> sorted([5, 2, 3, 1, 4])
[1, 2, 3, 4, 5]

>>> bin(999)
'0b1111100111'

>>> mylist = ['A', 'B' ,'C', 'D']
>>> enumerate(mylist)
<enumerate object at 0x0329F288>

>>>input(“Enter your name:”)
Enter your name:Sudhamshu B N

>>> oct(10)
'0o12'

>>> staticmethod(max([1,2,3,4]))
<staticmethod object at 0x03409208>

>>> bool(1)
True
>>> bool(0)
False

>>> expression = input("Enter an expression in terms of variable a:")
Enter an expression in terms of variable a:a*a^7+3*a+4
>>> a = int(input("Enter the value of a:"))
```

```
Enter the value of a:5
>>> eval(expression)
3

>>> int("1234")
1234

>>>f = open(“assignment.txt”, “r”)
>>>print(f.read())
This file is created for the purpose of assignment 4

>>> str(1234)
'1234'

>>> x = 10
>>> y = ‘Hi’
>>> z = ‘Hello’
>>> print(y)
Hi
>>> breakpoint()
--Return--
><stdin>(1)<module> () ->None
 (Pdb)
 (Pdb) print(z)
Hello
 (Pdb)
Hello

>>> x = 9
>>> exec ('print(5*x)')
45

>>>isinstance(5, int)
True

>>>ord(a)
97

>>> sum([1,2,3,4,5,6,7,8])
36

>>> bytearray("Sudhamshu", 'utf-8')
bytearray(b'Sudhamshu')
```

```
>>> def myFunc(x):
...   if x < 18:
...     return False
...   else:
...     return True
...
>>> adults = filter(myFunc, ages)
>>>
>>> for x in adults:
...   print(x)
...
18
24
32

>>> class myAge:
...   age = 18
...
>>> class myObj(myAge):
...   name = "Sudhamshu B N"
...   age = myAge
...
>>> x = issubclass(myObj, myAge)
>>> x
True

>>> pow(4,2)
16

>>> class Parent:
...   def __init__(self, txt):
...     self.message = txt
...   def printmessage(self):
...     print(self.message)
...
>>> class Child(Parent):
...   def __init__(self, txt):
...     super().__init__(txt)                    # super()
...
>>> x = Child("Hello, Sudhamshu here!")
>>> x.printmessage()
Hello, Sudhamshu here!
```

```
>>> bytes(4)
b'\x00\x00\x00\x00'

>>> float(1)
1.0

>>> iter(['a','e','i','o','u'])
<list_iterator object at 0x036E9880>

>>>print(“Hello, World!”)
Hello, World!

>>> num=25
>>> callable(num)
False

>>> txt = "Student studies at {DSCE_in:.2f} Year !"
>>> print(txt.format(DSCE_in = 3))
Student studies at 3.00 Year !

>>> len([1,2,3,4,5,6])
6

>>> class person:
...     def __init__(self):
...         self.__name=''
...     def setname(self, name):
...         print('setname() called')
...         self.__name=name
...     def getname(self):
...         print('getname() called')
...         return self.__name
...     name=property(getname, setname)
…

>>> a = 5 + 7j
>>> type(a)
<class 'complex'>

>>> chr(101)
'e'

>>> list((1,2,3,4,5))
[1, 2, 3, 4, 5]
```

```
>>> for i in range(3):
...     print("**")
...
**
**
**

>>> mylist = ['apple', 'banana', 'cherry']
>>> x = frozenset(mylist)
>>> x
frozenset({'apple', 'banana', 'cherry'})

>>> class Person:
...     name="Sudhamshu"
...     age="19"
...     country="India"
...
>>> vars(Person)
mappingproxy({'__module__': '__main__', 'name': 'Sudhamshu', 'age': '19', 'country': 'India',
'__dict__': <attribute '__dict__' of 'Person' objects>, '__weakref__': <attribute '__weakref__' of
'Person' objects>, '__doc__': None})

>>> def localsPresent():
...     present = True
...     print(present)
...     locals()['present'] = False;
...     print(present)
...
>>> localsPresent()
True
True

>>> x='foo'
>>> repr(x)
"'foo'"

>>> roll_no = [ 4, 1, 3, 2 ]
>>> marks = [ 40, 50, 60, 70 ]
>>> name=["Sudhamshu","Nikhil","Yogesh","Ramesh"]
>>> zip(name, roll_no, marks)
<zip object at 0x036EAD88>
```

```
>>> x = compile('print(55)', 'test', 'eval')
>>> exec(x)
55

>>> age = 23
>>>
>>> globals()['age'] = 25
>>> print('The age is:', age)
The age is: 25

>>> def myfunc(n):
...   return len(n)
...
>>> x = map(myfunc, ('apple', 'banana', 'cherry'))
>>> x
<map object at 0x036E9A30>

>>> a=["a","b","c"]
>>> b=reversed(a)
>>> for i in b:
...     print(i)
...
c
b
a

>>> import os
>>> os.path
<module 'ntpath' from 'C:\\Users\\sudha\\AppData\\Local\\Programs\\Python\\Python38-32\\lib\\ntpath.py'>
>>> os.name
'nt'

>>> complex(9,5)
(9+5j)

>>> max([1,2,3,4,5,6,7,8])
8

>>> round(5.6)
6
>>> round(5.3)
5
```