In [1]:

```
square = lambda a:a**a
result=square(2)
print(result)
```

```
4
```

In [1]:

```
mul = lambda a,b:a*b
result = mul(5,3)
print(result)
```

```
15
```

In [4]:

```
squre = lambda a:a*a
result=square(6)
print(result)
```

```
46656
```

In [5]:

```
six = lambda:6
result = six()
print(result)
```

```
6
```

In [6]:

```
factorial = lambda a: a*factorial(a-1) if (a>1) else 1
result = factorial(5)
print(result)
```

```
120
```

In [7]:

```
tuple = [ 'a','anil','kumar','iam','b']
print('a' in tuple)
print(tuple[1])
```

```
True
anil
```

In [8]:

```
tuple = ['1','a','b','c','d',2,3,4,'t']
del tuple[0]
print(tuple)
```

```
['a', 'b', 'c', 'd', 2, 3, 4, 't']
```

In [9]:

```
tuple=12,"iam","@@@@"
print(tuple)
```

```
(12, 'iam', '@@@@')
```

In [10]:

```
tuple=[12,(12,"iam",13,"here"),[122,"hlo..."]]
print(tuple)
```

```
[12, (12, 'iam', 13, 'here'), [122, 'hlo...']]
```

In [11]:

```
tuple = [12,"hiiii",13,"how are you!!!!","@@@@"]
print(tuple)
```

```
[12, 'hiiii', 13, 'how are you!!!!', '@@@@']
```

In [12]:

```
tuple =[12,13,13,12.5,22]
for x in tuple :
    print(x)
```

```
12
13
13
12.5
22
```

In [13]:

```
tuple =[12,13,13,12.5,22]
print(tuple[2])
print(tuple[0])
```

```
13
12
```

In [14]:

```
import re
txt = "iam here good"
x= re.split("\s",txt)
print(x)
```

```
['iam', 'here', 'good']
```

In [15]:

```
#swapping of two numbers
num1 = input('enter a number1:')
num2 = input('enter a number2:')
temp=num1
num1=num2
num2=temp
print('the value of num1 after swapping:{}'.format(num1))
print('the value of num2 after swapping:{}'.format(num2))
```

```
enter a number1:1
enter a number2:2
the value of num1 after swapping:2
the value of num2 after swapping:1
```

In [18]:

```
num1=input('Enter a number:')
num2=input('Enter a number:')
sum = num1+num2
print('the sum of {0} and {1} is {2}'.format('num1','num2',sum))
```

```
Enter a number:1
Enter a number:2
the sum of num1 and num2 is 12
```

In [19]:

```
import re
txt = '{"iam":"anil"}'
rr=re.loads(txt)
print(rr)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
C:\Users\ANILKU~1\AppData\Local\Temp/ipykernel_17520/2285070252.py in <modul
e>
      1 import re
      2 txt = '{"iam":"anil"}'
----> 3 rr=re.loads(txt)
      4 print(rr)

AttributeError: module 're' has no attribute 'loads'
```

In [20]:

```
import jason
x={"name":"SHIVA","age":22,"id":1020,"car":"mmm"}
y=jason.dumps(x)
print(y)
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
C:\Users\ANILKU~1\AppData\Local\Temp/ipykernel_17520/2500328919.py in <modul
e>
----> 1 import jason
      2 x={"name":"SHIVA","age":22,"id":1020,"car":"mmm"}
      3 y=jason.dumps(x)
      4 print(y)

ModuleNotFoundError: No module named 'jason'
```

In [21]:

```
def bubblesort(list):
for iter_num in range(len(list)-1,0,-1):
    for idx in range(iter_num):
            if list[idx]>list[idx+1]:
                temp =list[idx]
                list[idx] = list[idx+1]
                list[idx+1]= temp
list = [19,2,31,45,15,22,18,19]
bubblesort(list)
print(list)
```

```
  File "C:\Users\ANILKU~1\AppData\Local\Temp/ipykernel_17520/2904774605.py",
line 2
    for iter_num in range(len(list)-1,0,-1):
    ^
IndentationError: expected an indented block
```

In [22]:

```
print("Hello")
a = "Hello"
print(a)
```

```
Hello
Hello
```

In [23]:

```
 a = "Hello, World!"
print(a[1])
```

```
e
```

In [24]:

```
 for x in "banana":
       print(x)
```

```
b
a
n
a
n
a
```

In [25]:

```
a = "Hello, World!"
print(len(a))
```

```
13
```

In [26]:

```
 txt = "The best things in life are free!"
if "free" in txt:
    print("Yes, 'free' is present.")
```

```
Yes, 'free' is present.
```

In [27]:

```
 b = "Hello, World!"
print(b[2:5])
```

```
llo
```

In [28]:

```
a = "Hello, World!"
print(a.upper())
```

```
HELLO, WORLD!
```

In [29]:

```
a = "Hello, World!"
print(a.lower())
```

```
hello, world!
```

In [30]:

```
a = " Hello, World! "
print(a.strip())
```

```
Hello, World!
```

In [31]:

```
a = "Hello, World!"
print(a.replace("H", "J"))
```

```
Jello, World!
```

In [32]:

```
a = "Hello, World!"
print(a.split(","))
```

```
['Hello', ' World!']
```

In [33]:

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

```
['apple', 'banana', 'cherry']
```

In [34]:

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

```
3
```

In [35]:

```
 thislist = ["apple", "banana", "cherry"]
print(thislist[1])
```

```
banana
```

In [36]:

```
 thislist = ["apple", "banana", "cherry", "orange", "kiwi", "melon", "mango"]
print(thislist[2:5])
```

```
['cherry', 'orange', 'kiwi']
```

In [37]:

```
thislist = ["apple", "banana", "cherry"]
if "apple" in thislist:
    print("Yes, 'apple' is in the fruits list")
```

```
Yes, 'apple' is in the fruits list
```

In [38]:

```
thislist = ["apple", "banana", "cherry"]
thislist.insert(2, "watermelon")
print(thislist)
```

```
['apple', 'banana', 'watermelon', 'cherry']
```

In [39]:

```
thislist = ["apple", "banana", "cherry"]
thislist.append("orange")
print(thislist)
```

```
['apple', 'banana', 'cherry', 'orange']
```

In [40]:

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

```
['apple', 'cherry']
```

In [41]:

```
thislist = ["apple", "banana", "cherry"]
thislist.pop(1)
print(thislist)
```

```
['apple', 'cherry']
```

In [42]:

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

```
['apple', 'banana']
```

In [43]:

```
thislist = ["apple", "banana", "cherry"]
del thislist
```

In [44]:

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

```
[]
```

In [45]:

```
thislist = ["apple", "banana", "cherry"]
for x in thislist:
 print(x)
```

```
apple
banana
cherry
```

In [46]:

```
thislist = ["apple", "banana", "cherry"]
for i in range(len(thislist)):
    print(thislist[i])
```

```
apple
banana
cherry
```

In [47]:

```
thislist = ["apple", "banana", "cherry"]
i = 0
while i < len(thislist):
 print(thislist[i])
 i = i + 1
```

```
apple
banana
cherry
```

In [48]:

```
print("\n")
thelist = ["apple", "banana", "cherry"]
[print(x) for x in thelist]
```

```


apple
banana
cherry
```

Out[48]:

```
[None, None, None]
```

In [49]:

```
for x in range(2, 30, 3):
 print(x)
```

```
2
5
8
11
14
17
20
23
26
29
```

In [ ]:

```
i = 1
while i < 6:
    print(i)
    i += 1
```

In [ ]:

```
def foo():
    pass
print(type(foo))
print("foo name attribute: ",foo.__name__)
```

In [ ]:

```
class Abc:
    def m1(self):
    pass
```

In [ ]:

```
class sample:
    def __init__(self):
        self.a = 13
        self.b = 15
s = sample()
print(s.a)
```

In [ ]:

```
class sample:
   def __call__(self):
     pass
s = sample()
print(callable(s))
```

In [ ]:

```
def local_function():
    print("This is a local function")
s = local_function()
```

In [ ]:

```
def abc(x):
    return x**2
def xyz(func):
    num=10
    return func(num)
xyz(abc)
```

In [ ]:

```
def addexclamation(function):
    def add():
        func = function()
    return func +" !!!"
    return add
def sentence():
    return "hello all"
msg = addexclamation(sentence)
print(msg())
```

In [ ]:

```
list1 = [x**2 for x in range(10)]
print(list1)
```

In [ ]:

```
lst = [2,5,6,3,8]
dict = {x:x+5 for x in lst if x%2==0}
print(dict)
```

In [ ]:

```
l1 = [1,2,3]
l2 = [5,8,7]
dict1 = {key:value for (key,value) in zip(l1, l2)}
print(dict1)
```

In [ ]:

```
dict = {'jack': 38, 'michael': 48, 'guido': 57, 'john': 33}
new_dict = {k: v for (k, v) in dict.items() if v%2!=0 if v<40}
print(new_dict)
```

In [ ]:

```
dct = {'jack': 38, 'michael': 48, 'guido': 57, 'john': 33}
new_dct = {k: ('old' if v > 40 else 'young')
for (k, v) in dct.items()}
print(new_dct)
```

In [ ]:

```
lst = [1,2,3,2,5,3]
set1 = {x for x in lst}
print(set1)
```

In [ ]:

```
nput_list = [1, 2, 3, 4, 4, 5, 6, 6, 6, 7, 7]
set2 = {var for var in input_list if var % 2 == 0}
print(set2)
```

In [ ]:

```
lst = [2,7,5,0,4,6]
gen = (x for x in lst if x%3==0)
for i in gen:
    print(i)
```

In [ ]:

```
gen2 = (i**2 for i in range(5))
for item in gen2:
    print(item)
```

In [ ]:

```

class Abc:
    pass
```

In [ ]:

```
class Values:
  def __init__(self):
       self.a = 13
        self.b = 15
s = Values()
print(s.a)
print(s.b)
s.b = 29
print(s.b)
```

In [ ]:

```
class Values2:
 def __init__(hi,a,b):
     hi.a = a
    hi.b = b
s = Values2(int(input()),int(input()))
print(s.a,s.b)
```

In [ ]:

```
class Student:
    def __init__(self,name,age):
        self.name = name
        self.age = age

 def printname(hi):
    print("my name is :"+hi.name)
S1 = Student("arathi",22)
print(S1.name)
print(S1.age)
S2 = Student("surya",20)
S2.printname()
```

In [ ]:

```
class Student1(Campus):
     pass
s= Student1(456,'ABC') #accessing inherited props of parent class
s.show()
```

In [ ]:

```

class Student2(Campus):

 def __init__(self,code,name,address):

 Campus.__init__(self,code,name)
        self.address = address

 def all3(self):
        print(self.code,self.name,self.address)

 st = Student2(156,'QWE','kerala')
st.show()
st.all3()
```

In [ ]:

```
class A:
  def __init__(self):
     self.str1="hi"
    print("A")
class B:
 def __init__(self):
    self.str2 = "hello"
    print("B")

 class C(A,B):
   def __init__(self):
     A.__init__(self)
        B.__init__(self)
self.str3 = "bye"
 print("C")
 def printstr(self):
    print(self.str1,self.str2,self.str3)

 cobj = C()
cobj.printstr()
```

In [ ]:

```
for city in ["Berlin", "Vienna", "Zurich"]:
print(city)
```

In [ ]:

```
def iterable(obj):
    try:
        iter(obj)
        return True

    except TypeError:
        return False
```

In [ ]:

```
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')

sum = float(num1) + float(num2)
print("sum of {0} and {1} is {2}".format(num1,num2,sum))
```

In [ ]:

```
y = input('Enter value of y: ')
temp = x
x = y
y = temp
print('The value of x after swapping: {}'.format(x))
print('The value of y after swapping: {}'.format(y))
```

In [ ]:

```
a = 33
b = 33
if b > a:
    print("b is greater than a")
elif a == b:
    print("a and b are equal")
```

In [ ]:

```
import json
x = {"name": "John","age": 30,"married": True,"divorced": False, "children": ("Ann","Billy"
"pets": None,"cars": [ {"model": "BMW 230", "mpg": 27.5},{"model": "Ford Edge", "mpg": 24.1
# convert into JSON:
y = json.dumps(x)
# the result is a JSON string:
print(y)
```

In [ ]:

```
a=5
b=6
c=7
s=(a+b+c)/2
# caluculate the area
area =(s*(s-a)*(s-b)*(s-c))**0.5
print('the area of triangle is %0.2f'%area)
```

In [ ]:

```
print("hiii")
```

In [ ]:

```

```