In [1]:

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

```
Hello
Hello
```

In [2]:

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

```
e
```

In [5]:

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

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

In [7]:

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

```
13
```

In [9]:

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

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

In [10]:

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

```
llo
```

In [11]:

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

```
HELLO, WORLD!
```

In [12]:

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

```
hello, world!
```

In [13]:

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

```
Hello, World!
```

In [14]:

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

```
Jello, World!
```

In [15]:

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

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

In [16]:

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

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

In [17]:

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

```
3
```

In [18]:

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

```
banana
```

In [19]:

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

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

In [21]:

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

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

In [22]:

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

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

In [23]:

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

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

In [24]:

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

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

In [25]:

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

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

In [26]:

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

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

In [27]:

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

In [28]:

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

```
[]
```

In [29]:

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

```
apple
banana
cherry
```

In [31]:

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

```
apple
banana
cherry
```

In [32]:

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

```
apple
banana
cherry
```

In [33]:

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

```
apple
banana
cherry
```

In [34]:

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

```
apple
banana
cherry
```

Out[34]:

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

In [35]:

```
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 [ ]:

```

y = lambda x:x*2
print(y(5))
#Eg2:
lst = [1,5,7,14]
newlst = list(filter(lambda a:a%7==0,lst))
print(newlst)
```

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 [ ]:

```

list2 = [5,2,7,8,14,22]
list3 = [x for x in list2 if x%2==0]
print(list3)
```

In [ ]:

```
x = lambda a:a**2
print(x(5))
```

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 College:
  def __init__(self): #constructor to initialize attributes
        print("Welcome")
s1 = College()
```

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: ')

# create a temporary variable and swap the values
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 [ ]:

```
thistuple = ("apple", "banana", "cherry")
print(thistuple)
```

In [ ]:

```
thisdict =  {"brand": "Ford", "model": "Mustang", "year": 1964}
print(thisdict)
```

In [ ]:

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

In [ ]:

```
x = lambda a, b, c: a + b + c
print(x(5, 6, 2))
```

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 [ ]:

```
tuple1 = (14, 52, 17, 24)
print(tuple1[1])
print(tuple1[3])
```

In [ ]:

```
tuple1 = (14, 52, 17, 24)
print( len(tuple1) )
```

In [ ]:

```
tuple1 = (14, 52, 17, 24)
for item in tuple1:
    print(item)
```

In [ ]:

```
tuple1 = (14, 52, 17, 24)

index = 0
while index<len(tuple1):
    print(tuple1[index])
    index = index + 1
```

In [ ]:

```
my_tuple = (1, 2, 3)
     print(my_tuple)
```

In [ ]:

```
my_tuple = (1, "Hello", 3.4)
print(my_tuple)
```

In [ ]:

```
my_tuple = ("mouse", [8, 4, 6], (1, 2, 3))
print(my_tuple)
```

In [ ]:

```
my_tuple = ("hello")
print(type(my_tuple))
```

In [ ]:

```
my_tuple = "hello"
print(type(my_tuple))
```

In [ ]:

```

```