In [11]:
```python
def show_stars(rows):
   for i in range(rows):
      for j in range(i+1):
         # printing stars
         print("*",end=" ")
      print("\r")

# take inputs
rows= int(input('Enter the number of rows: '))

# calling function
show_stars(rows)
```

```
Enter the number of rows: 5
*
* *
* * *
* * * *
* * * * *
```

Removal of string at the mentioned position

In [12]:
```python
test_str ="Artificial Intelligence"

# Printing original string
print ("The original string is : " , test_str)

# Removing char at pos 7
# using for loop
new_str = ""

for i in range(len(test_str)):
    if i != 6:
        new_str = new_str + test_str[i]

# Printing string after removal
print ("The string after removal of i'th character : " ,new_str)
```

```
The original string is :  Artificial Intelligence
The string after removal of i'th character :  Artifiial Intelligence
```

printing of numbers which are divisible by 5

In [13]:
```python
lower = int(input("Enter lower range limit:"))
upper = int(input("Enter upper range limit:"))
for i in range(lower, upper+1):
   if (i%5==0):
      print(i)
```

```
Enter lower range limit:10
Enter upper range limit:99
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
```

counting the substring

In [14]:
```python
string = 'Hi! everyone welcome to python'
sub_string = 'Hi'
results = 0
sub_len = len(sub_string)
for i in range(len(string)):
    if string[i:i+sub_len] == sub_string:
        results += 1
print (results)
```

```
1
```

printing numbers in pyramid

In [40]:
```python
def numbers(rows):
   for i in range(rows+1):
      for j in range(i):
         # printing numbers
         print(i,end=" ")
      print("\r")

# take inputs
rows= int(input('Enter the number of rows: '))

# calling function
numbers(rows)
```

```
Enter the number of rows: 5

1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

lists in python

program to interchange first and last elements in a list

In [43]:
```python
mic = [21, 7, 33, 57, 23, 87]
print("Initial List : ", myList)

# finding the length of list
length = len(myList)

# Swapping first and last element
temp = myList[0]
myList[0] = myList[length - 1]
myList[length - 1] = temp

print("List after Swapping : ", myList)
```

```
Initial List :  [87, 7, 33, 57, 23, 21]
List after Swapping :  [21, 7, 33, 57, 23, 87]
```

swaping the elements in the list

In [47]:
```python
def swaping(sl,pos1,pos2):
    n = len(sl)
    # Swapping
    temp = sl[pos1]
    sl[pos1] = sl[pos2]
    sl[pos2] = temp
    return sl

k= [12,34,56,78,90]
pos1= 3
pos2= 5

print(l)
print("Swapped list: ",swaping(k,pos1-1,pos2-1))
```

```
[12, 34, 90, 78, 56]
Swapped list:  [12, 34, 90, 78, 56]
```

to find length of list

In [52]:
```python
list1 = ["Hello", "everyone", 1, 2, 3]
list2=["mango","gauva",'##',8]
print ("Number of items in the list1 = ", len(list1))
print ("Number of items in the list2 = ", len(list2))
```

```
Number of items in the list1 =  5
Number of items in the list2 =  4
```

Maximum of two numbers

In [53]:
```python
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))

# printing the maximum value
if(num1 > num2):
  print(num1, "is greater")
elif(num1 < num2):
    print(num2, "is greater")
else:
    print("Both are equal")
```

```
Enter the first number: 78
Enter the second number: 43
78 is greater
```

minimum of two numbers

In [54]:
```python
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))

# printing the minimum value
if(num1 < num2):
  print(num1, "is lowest")
elif(num2 < num1):
    print(num2, "is lowest")
else:
    print("Both are equal")
```

```
Enter the first number: 56
Enter the second number: 89
56 is lowest
```

tuples in python

size of tuples

In [16]:
```python
import sys

# sample Tuple
fly= ("hlo", 1, "welcome", 2, "to", "python",3)
Jass = ("mango", "is", "my", "favourite", "fruit", "banana")
Ross = ((1, "Lion"), ( 2, "Tiger"), (3, "Fox"), (4, "Wolf"))

# print the sizes of sample Tuples
print("Size of Tuple1: " , str(sys.getsizeof(fly)) , "bytes")
print("Size of Tuple2: " , str(sys.getsizeof(Jass)) , "bytes")
print("Size of Tuple3: " , str(sys.getsizeof(Ross)) , "bytes")
```

```
Size of Tuple1:  96 bytes
Size of Tuple2:  88 bytes
Size of Tuple3:  72 bytes
```

Maximum and minimum k element in tuple

In [20]:
```python
sample_tuple = (98, 99, 0, 9, 10, 8)

print("The tuple is : ")
print(sample_tuple)

K = 2
print("The value of K has been initialized to ")
print(K)
my_result = []
sample_typle = list(sample_tuple)
temp = sorted(sample_tuple)

for idx, val in enumerate(temp):
   if idx < K or idx >= len(temp) - K:
      my_result.append(val)
my_result = tuple(my_result)

print("The result is : " )
print(my_result)
```

```
The tuple is :
(98, 99, 0, 9, 10, 8)
The value of K has been initialized to
2
The result is :
(0, 8, 98, 99)
```

sum of tuple elements

In [26]:
```python
myTuple = (7, 8, 9, 1, 99, 7)

# printing original tuple
print("The original tuple is : " , str(myTuple))

# finding sum of all tuple elements
tupSum = sum(list(myTuple))

# Printing sum of tuple elements
print("The sum of tuple elements are : " , str(tupSum))
```

```
The original tuple is :  (7, 8, 9, 1, 99, 7)
The sum of tuple elements are :  131
```

In [25]:
```python
# create a tuple
t = (1, 2, 5, 4)
# sum of tuple elements
print(sum(t))
```

```
12
```

Strings in python

program to check whether the string is Symmetrical or Palindrome

In [28]:
```python
# function for palindrome
def palin(string):

  # declare and initialize with the starting and ending indexes
  st = 0
  end = len(string)-1
  f = 0

  while(st<end):

    if (string[st]== string[end]):

      st += 1
      end -= 1

    else:
      f = 1
      break;
  if f == 0:
    print("The entered string is palindrome")
  else:
    print("The entered string is not palindrome")

# symm function to check string symmetrical or not
def symm(string):

  l = len(string)
  flag = 0

  # to check length of string even or odd
  # to calculate middle value accordingly
  if l%2 == 0:
    mid = l//2 # for even length
  else:
    mid = l//2 + 1 # for odd length

  s1 = 0  # starting for first portion of string
  s2 = mid # starting for rest portion of string after middle value

  while(s1 < mid and s2 < l):

    if (string[s1] == string[s2]): # comparing from start of both portions
                                   # of given string
      s1 = s1 + 1
      s2 = s2 + 1
    else:
      flag = 1
      break


  if flag == 0:
    print("The entered string is symmetrical")
  else:
    print("The entered string is not symmetrical")


string = input("Enter the string: ")
palin(string)
symm(string)
```

```
Enter the string: madam
The entered string is palindrome
The entered string is not symmetrical
```

Reversing the words in a given string

In [31]:
```python
x = input("Enter any string: ")
#take input from user
a = x.split()
#use split method to split at whitespaces
a.reverse()
#reverse all the elements of the string
print(' '.join(a))
#concatenate them into a string
```

```
Enter any string: welcome to artificial intellgence
intellgence artificial to welcome
```

remove i'th character from string

In [34]:
```python
def remove_char(s, i):

    for j in range(len(s)):
        if j==i:
            s=s.replace(s[i],"",1)
    return s

string = "Engineering"
# Remove i-th index element
i = 4
print(remove_char(string,i-1))
```

```
Engneering
```

Sets in python

size of sets

In [36]:
```python
import sys

# sample Sets
Set1 = {"A", 1, "B", 2, "C", 3}
Set2 = {"mango", "banana", "gauva", "jackfruit", "papaya", "kosaimasse"}
Set3 = {(1, "python"), ( 2, "c"), (3, "c++")}

# print the sizes of sample Sets
print("Size of Set1: " , str(sys.getsizeof(Set1)) , "bytes")
print("Size of Set2: " , str(sys.getsizeof(Set2)) , "bytes")
print("Size of Set3: " , str(sys.getsizeof(Set3)) , "bytes")
```

```
Size of Set1:  472 bytes
Size of Set2:  472 bytes
Size of Set3:  216 bytes
```

Iterate over a set

In [39]:
```python
#Create a set
num_set = set([0, 1, 2, 0, 11, 99])
for n in num_set:
  print(n, end=' ')
print("\n\nCreating a set using string:")
char_set = set("interpreter")
# Iterating using for loop
for val in char_set:
    print(val, end=' ')
```

```
0 1 2 99 11

Creating a set using string:
i r e p t n
```

Dictionaries in Python

sorting of dictionary by its key value

In [56]:
```python
dict = {6:'Murugan' ,2:'sandhiya' ,1:'peter' ,9:'Micheal' ,7:'TEsla' ,8:'thomas' }

key_sort= sorted(dict.keys())
print("Sorted keys",key_sort)

value_sort= sorted(dict.items())
print("Sorted Values",value_sort)
```

```
Sorted keys [1, 2, 6, 7, 8, 9]
Sorted Values [(1, 'peter'), (2, 'sandhiya'), (6, 'Murugan'), (7, 'TEsla'), (8, 'thomas'), (9, 'Micheal')]
```

keys having multiple inputs

In [57]:
```python
my_dict = {}

a, b, c = 15, 26, 38
my_dict[a, b, c] = a + b - c

a, b, c = 5, 4, 11
my_dict[a, b, c] = a + b - c

print("The dictionary is :")
print(my_dict)
```

```
The dictionary is :
{(15, 26, 38): 3, (5, 4, 11): -2}
```

Matrices in Python

Python – Assigning Subsequent Rows to Matrix first row elements

In [61]:
```python
trial_list = [[5, 0, 99], [2, 6, 9], [7, 4, 2], [0, 3, 9]]

# printing original list
print("The original list : " , str(trial_list))

# pairing each 1st col with next rows in Matrix
res = {trial_list[0][ele] :  trial_list[ele + 1] for ele in range(len(trial_list) - 1)}

# printing result
print("The Assigned Matrix : " , str(res))
```

```
The original list :  [[5, 0, 99], [2, 6, 9], [7, 4, 2], [0, 3, 9]]
The Assigned Matrix :  {5: [2, 6, 9], 0: [7, 4, 2], 99: [0, 3, 9]}
```

Adding and Subtracting Matrices in Python

In [64]:
```python
import numpy

# Matrix 1
A=[ [1, 8, 99], [3, 84, 5], [0, 7, 8] ]

# Matrix 2
B=[ [7, 6, 98], [1, 0, 3], [5, 3, 9] ]

print("Result: ")
print(numpy.add(A,B))
```

```
Result:
[[  8  14 197]
 [  4  84   8]
 [  5  10  17]]
```

In [67]:
```python
import numpy

# Matrix 1
A=[ [1, 8, 99], [3, 84, 5], [0, 7, 8] ]

# Matrix 2
B=[ [7, 6, 98], [1, 0, 3], [5, 3, 9] ]

print("Result: ")
print(numpy.subtract(A,B))
```

```
Result:
[[-6  2  1]
 [ 2 84  2]
 [-5  4 -1]]
```

Functions in python

In [70]:
```python
def fun(arg1, arg2):
    return arg1**arg2

# import required modules
import inspect

# use signature()
print(inspect.signature(fun))
```

```
(arg1, arg2)
```

Printing Multiple Arguments in Python

In [73]:
```python
def sample(name, num="25"):
    print("Hello from", name + ', ' + num)



sample("python")
sample("python", "26")
```

```
Hello from python, 25
Hello from python, 26
```