Python basic programs

for
if
while
even
odd
n num
compare a,b
compa a,b,c

In [1]:
```
# simple if
a=int(input("enter a value"))
if a>=0:
    print("entered num {} positive".format(a))
```

```
enter a value11
entered num 11 positive
```

In [2]:
```
# if-elif
b=int(input("enter b value"))
if b>0:
    print("entered num {} is possitive".format(b))
elif b<0:
    print("entered num {} is negative ".format(b))
else:
    print("entered num is zero")
```

```
enter b value-1
entered num -1 is negative
```

In [3]:
```
a=int(input("enter a value: "))
b=int(input("enter b value: "))
if a>b:
    print("in entered numbers {} is grater than {}".format(a,b))
elif a<b:
    print("in entered numbers {} is less than {} or {} is
grater".format(a,b,b))
else:
    print("entered numbers both are equal")
```

```
enter a value: 22
enter b value: 12
in entered numbers 22 is grater than 12
```

In [4]:
```
#a, b c comparision
a=int(input("enter a value: "))
b=int(input("enter b value: "))
c=int(input("enter c value: "))
print("entered values are a={},b={},c{}".format(a,b,c))
if a>b:
    if a>c:
        print("{} is grater".format(a))
    else:
        print("{} is grater".format(c))
else:
    if b>c:
        print("{} is grater".format(b))
    else:
        print("{} is grater".format(c))
```

```
enter a value: 3
enter b value: 6
enter c value: 3
entered values are a=3,b=6,c3
6 is grater
```

In [5]:
```
# N numbers
n=int(input("enter n value: "))
i=0
while i<=n:
    print(i)
    i+=1
```

```
enter n value: 12
0
1
2
3
4
5
6
7
8
9
10
11
12
```

In [6]:
```
# even numbers
n=int(input("enter n value"))
for i in range(n):
    if i%2==0:
        print(i)
```

```
enter n value10
0
2
4
6
8
```

In [7]:
```
# odd numbers
n=int(input("enter n value"))
for i in range(n):
    if i%2!=0:
        print(i)
```

```
enter n value12
1
3
5
7
9
11
```

In [8]:
```
for i in range(1,20,2):
    print(i)
```

```
1
3
5
7
9
11
13
15
17
19
```

In [9]:
```
name="shekhar"
for i in name:
    print(i)
```

```
s
h
e
k
h
a
r
```

In [10]:
```
l=[2,4,6,7,55,9,44]
for i in range(len(l)):
    print(l[i])
```

```
2
4
6
7
55
9
44
```

In [11]:
```
for i in range(5):
    for j in range(5):
        print("*",end=" ")
    print(" ")
```

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

In [12]:
```
for i in range(5):
    for j in range(5):
        if i>j:
            print("*",end=" ")
    print(" ")
```

```
*
* *
* * *
* * * *
```

In [13]:
```
for i in range(5):
    for j in range(5):
        if i<j:
            print("*",end=" ")
    print(" ")
```

```
* * * *
* * *
* *
*
```

In [14]:
```
for i in range(10):
    if i==5:
        break
    print(i)
```

```
0
1
2
3
4
```

In [15]:
```
for i in range(10):
    if i==4:
        continue
    print(i)
```

```
0
1
2
3
5
6
7
8
9
```

In [16]:
```
i=1
while i<=10:
    if i==5:
        pass
    print(i)
    i+=1
```

```
1
2
3
4
5
6
7
8
9
10
```

In [17]:
```
# global function
x=input()
def myfunc():
    print("python is "+x)
```

```
12
```

In [18]:
```
myfunc()
```

```
python is 12
```

In [19]:
```
x="haiii"
def my_func():
    x="bye"
    print("thi is local python is "+x)
my_func()
print("tihis is golbal python is "+x)
```

```
thi is local python is bye
tihis is golbal python is haiii
```

In [20]:
```
x = "awesome"

def myfunc():
  global x
  x = "fantastic"

myfunc()

print(" this is global but we can choose local as global by using global keyword Python is " + x)
```

```
 this is global but we can choose local as global by using global keyword Pytho
n is fantastic
```

Math function

In [21]:
```
import math
```

In [22]:
```
math.ceil(10.3) # to the next num
```

Out[22]:
```
11
```

In [23]:
```
math.floor(10.3)
```

Out[23]:
```
10
```

In [24]:
```
a=10
b=-20
math.copysign(a,b)
```

Out[24]:
```
-10.0
```

In [25]:
```
math.fabs(a)
```

Out[25]:
```
10.0
```

In [26]:
```
math.fabs(b)
```

Out[26]:
```
20.0
```

In [27]:
```
math.factorial(5)
```

Out[27]: 120

In [28]:
```
math.factorial(6)
```

Out[28]: 720

In [29]:
```
l=[2,4,6,7,55,9,44]
math.fsum(l)
```

Out[29]: 127.0

In [30]:
```
math.gcd(4,5,6,7)
```

Out[30]: 1

In [31]:
```
math.gcd(12,3,4)
```

Out[31]: 1

In [32]:
```
math.gcd(2,4,6,8,10)
```

Out[32]: 2

In [33]:
```
math.lcm(2,3,4)
```

Out[33]: 12

In [34]:
```
math.pow(12,2)
```

Out[34]: 144.0

In [35]:
```
math.pow(2,10)
```

Out[35]: 1024.0

In [36]:
```
math.sqrt(100)
```

Out[36]: 10.0

In [37]:
```
math.sqrt(50)
```

Out[37]: 7.0710678118654755

In [38]:
```
math.sin(90)
```

Out[38]: 0.8939966636005579

In [39]:
```
math.cos(90)
```

Out[39]: -0.4480736161291701

In [40]:
```
math.sin(math.pi/2)
```

Out[40]: 1.0

In [41]:
```
math.cos(math.pi/2)
```

Out[41]: 6.123233995736766e-17

In [42]:
```
math.pi
```

Out[42]: 3.141592653589793

In [43]:
```
math.e
```

Out[43]: 2.718281828459045

In [44]:
```
math.tau
```

Out[44]: 6.283185307179586

In [45]:
```
math.inf
```

Out[45]: inf

In [46]:
```
a=math.nan
```

In [47]:
```
a
```

Out[47]: nan

Random functions

In [48]:
```
import random
```

In [49]:
```
l=[23,22,12,56,47,85,69,56,54,85,25,63,65,45,22]
```

In [50]:
```
random.choice(l)
```

Out[50]: 47

In [51]:
```
random.choices(l,k=4)
```

Out[51]: [56, 85, 69, 22]

In [52]:
```
random.randrange(10,15,3)
```

Out[52]: 10

In [53]:
```
random.random()
```

Out[53]:
```
0.11862785376612095
```

In [54]:
```
random.shuffle(l)
```

In [55]:
```
l
```

Out[55]:
```
[69, 63, 56, 45, 56, 23, 85, 47, 22, 85, 25, 22, 65, 54, 12]
```

In [56]:
```
random.uniform(10,20)
```

Out[56]:
```
14.592792585613939
```

list

accep all data types,ordered, changeble,allow duplicates,indexed,

append(),extend(),insert(),index()
remove(),pop(),del ,clear()
sort(),reverse(),copy()

In [57]:
```
l=["a","b","c","d"]
l1=["a",4,5.6,"hai",True]
```

In [58]:
```
#Length
len(l1)
```

Out[58]:
```
5
```

In [59]:
```
# type
type(l)
```

Out[59]:
```
list
```

In [60]:
```
# list constructor
a=list(("a",4,5.6,"hai",True))
```

In [61]:
```
type(a)
```

Out[61]:
```
list
```

In [62]:
```
len(a)
```

Out[62]:
```
5
```

In [63]:
```
# Access Items
l1[2]
```

Out[63]: 5.6

In [64]:
```
l1[2:]
```

Out[64]: [5.6, 'hai', True]

In [65]:
```
l1[:2]
```

Out[65]: ['a', 4]

In [66]:
```
l1[:-1]
```

Out[66]: ['a', 4, 5.6, 'hai']

In [67]:
```
l1[-1:]
```

Out[67]: [True]

In [68]:
```
l1[-3:-2]
```

Out[68]: [5.6]

In [69]:
```
l1=["a",4,5.6,"hai",True]
if 4 in l1:
    print("True")
else:
    print("False")
```

True

In [70]:
```
# change element
l1=["a",4,5.6,"hai",True]
l1[1]=32
l1
```

Out[70]: ['a', 32, 5.6, 'hai', True]

In [71]:
```
l1[0:3]=[4,5,6]
l1
```

Out[71]: [4, 5, 6, 'hai', True]

In [72]:
```
l1[1:3]=["hai"]
```

In [73]:
```
l1[-1]
```

Out[73]: True

In [74]:
```
# add elements to the list
l1.insert(2,"bye")
print(l1)
```

```
[4, 'hai', 'bye', 'hai', True]
```

In [75]:
```
l1=l1
lx=[3,45,88]
for x in l1:
    lx.append(x)
```

In [76]:
```
lx
```

Out[76]:
```
[3, 45, 88, 4, 'hai', 'bye', 'hai', True]
```

In [77]:
```
ln=l1+lx
```

In [78]:
```
ln
```

Out[78]:
```
[4, 'hai', 'bye', 'hai', True, 3, 45, 88, 4, 'hai', 'bye', 'hai', True]
```

In [79]:
```
lx.extend(l1)
```

In [80]:
```
lx.append(l1)
```

In [81]:
```
if "a" in lx:
    print("yes")
else:
    print("no")
```

```
no
```

In [82]:
```
# remove elements for list
l1=[2,3,4,5]
l2=[7,8,9,4]
l1.remove(3)
```

In [83]:
```
l1
```

Out[83]:
```
[2, 4, 5]
```

In [84]:
```
l1.remove(4)
```

In [85]:
```
l1
```

Out[85]:
```
[2, 5]
```

In [86]:
```
l2.pop()
```

Out[86]: 4

In [87]:
```
l2.pop(1)
```

Out[87]:
```
8
```

In [88]:
```
l2
```

Out[88]:
```
[7, 9]
```

In [89]:
```
del l2[0]
```

In [90]:
```
l2
```

Out[90]:
```
[9]
```

In [91]:
```
l=[]
for i in range(5,10,1):
   l.append(i)
```

In [92]:
```
l
```

Out[92]:
```
[5, 6, 7, 8, 9]
```

In [93]:
```
l2.extend(l)
```

In [94]:
```
l2
```

Out[94]:
```
[9, 5, 6, 7, 8, 9]
```

In [95]:
```
del l2[2]
```

In [96]:
```
del l
```

In [97]:
```
l2.clear()
```

In [98]:
```
l2
```

Out[98]:
```
[]
```

In [99]:
```
# loop lists
```

In [100]:
```
l1=list(range(5,20,2))
l2=list(range(10,30,3))
print("list one elements: ",l1)
print("list two elements: ",l2)
```

```
list one elements:  [5, 7, 9, 11, 13, 15, 17, 19]
list two elements:  [10, 13, 16, 19, 22, 25, 28]
```

In [101]:
```
# loop through list
for x in l1:
    print(x)
```

```
5
7
9
11
13
15
17
19
```

In [102]:
```
# loop through index
for i in range(len(l1)):
    print(l1[i])
```

```
5
7
9
11
13
15
17
19
```

In [103]:
```
# using while loop
l1=[5, 7, 9, 11, 13, 15, 17, 19]
i=0
while i<len(l1):
    print(l1[i])
    i+=1
```

```
5
7
9
11
13
15
17
19
```

In [104]:
```
# list comprehension
[print(x) for x in l1]
```

```
5
7
9
11
13
15
17
19
```

Out[104]:
```
[None, None, None, None, None, None, None, None]
```

In [105]:
```
newlist=[x for x in l1 if 11 in l1]
newlist
```

Out[105]:
```
[5, 7, 9, 11, 13, 15, 17, 19]
```

In [106]:
```
newlist=[x for x in l1 if x!=19 ]
newlist
```

Out[106]:
```
[5, 7, 9, 11, 13, 15, 17]
```

In [107]:
```
newlist=[x for x in range(10)]
newlist
```

Out[107]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [108]:
```
newlist=[x for x in range(10) if x<6]
newlist
```

Out[108]:
```
[0, 1, 2, 3, 4, 5]
```

In [109]:
```
newlist=[x for x in range(10,50,6)]
newlist=[11 for x in newlist]
newlist
```

Out[109]:
```
[11, 11, 11, 11, 11, 11, 11]
```

In [110]:
```
newlist=[x for x in range(5,50,7)]
newlist
```

Out[110]:
```
[5, 12, 19, 26, 33, 40, 47]
```

In [111]:
```
newlist=[5, 12, 19, 26, 33, 40, 47]
newlist=[x if x!=13 else 12 for x in newlist]
newlist
```

Out[111]:
```
[5, 12, 19, 26, 33, 40, 47]
```

In [112]:
```
# sort elements
l2=[2,3,4,5,6,1,2,39,5,1,23,6,5,47,45]
l2.sort()
```

In [113]:
```
l2.sort(reverse=True)
```

In [114]:
```
l2
```

Out[114]:
```
[47, 45, 39, 23, 6, 6, 5, 5, 5, 4, 3, 2, 2, 1, 1]
```

In [115]:
```
l=["SFGHDJfgh","fghjFFGH","fGHBJhjk"]
```

In [116]:
```
l.sort(key=str.lower)
```

In [117]:
```
l2[-15:-1]
```

Out[117]:
```
[47, 45, 39, 23, 6, 6, 5, 5, 5, 4, 3, 2, 2, 1]
```

In [118]:
```
l2.copy()
```

Out[118]:
```
[47, 45, 39, 23, 6, 6, 5, 5, 5, 4, 3, 2, 2, 1, 1]
```

In [119]:
```
l3=list(l2)
```

In [120]:
```
l3
```

Out[120]:
```
[47, 45, 39, 23, 6, 6, 5, 5, 5, 4, 3, 2, 2, 1, 1]
```

In [121]:
```
# join
l3=l1+l2
```

In [122]:
```
l3
```

Out[122]:
```
[5, 7, 9, 11, 13, 15, 17, 19, 47, 45, 39, 23, 6, 6, 5, 5, 5, 4, 3, 2, 2, 1, 1]
```

In [123]:
```
l1.append(l2)
```

In [124]:
```
l1
```

Out[124]:
```
[5,
 7,
 9,
 11,
 13,
 15,
 17,
 19,
 [47, 45, 39, 23, 6, 6, 5, 5, 5, 4, 3, 2, 2, 1, 1]]
```

In [125]:
```
l1.extend(l2)
```

In [126]:
```
l1
```

Out[126]:
```
[5,
 7,
 9,
 11,
 13,
 15,
 17,
 19,
 [47, 45, 39, 23, 6, 6, 5, 5, 5, 4, 3, 2, 2, 1, 1],
 47,
 45,
 39,
 23,
 6,
 6,
 5,
 5,
 5,
 4,
 3,
 2,
 2,
 1,
 1]
```

Tuple

accep all data types,ordered, unchangeble,allow duplicates,indexed

same as list
count(),index()

In [127]:
```
# create tuple
t=(1,2,3,4,5)
```

In [128]:
```
t
```

Out[128]:
```
(1, 2, 3, 4, 5)
```

In [129]:
```
type(t)
```

Out[129]:
```
tuple
```

In [130]:
```
len(t)
```

Out[130]:
```
5
```

In [131]:
```
# tuple with single element
t=("hai")
```

In [132]:
```
t
```

Out[132]: 'hai'

In [133]:
```
type(t)
```

Out[133]: str

In [134]:
```
t=("haii",)
```

In [135]:
```
type(t)
```

Out[135]: tuple

In [136]:
```
t=("apple",56,True)
```

In [137]:
```
t
```

Out[137]: ('apple', 56, True)

In [138]:
```
# tuple constructure
t=((4,"haiii",False))
```

In [139]:
```
type(t)
```

Out[139]: tuple

In [140]:
```
len(t)
```

Out[140]: 3

In [141]:
```
# access tuple elements
```

In [142]:
```
t=tuple(range(5,60,6))
```

In [143]:
```
t
```

Out[143]: (5, 11, 17, 23, 29, 35, 41, 47, 53, 59)

In [144]:
```
type(t)
```

Out[144]: tuple

In [145]:
```
t[4]
```

Out[145]: 29

In [146]:
```
print([x for x in range(len(t))])
print(t)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
(5, 11, 17, 23, 29, 35, 41, 47, 53, 59)
```

In [147]:
```
t[5:]
```

Out[147]:
```
(35, 41, 47, 53, 59)
```

In [148]:
```
t[:2]
```

Out[148]:
```
(5, 11)
```

In [149]:
```
t[-1]
```

Out[149]:
```
59
```

In [150]:
```
t[2:4]
```

Out[150]:
```
(17, 23)
```

In [151]:
```
if 6 in t:
    print("yes")
else:
    print("no")
```

```
no
```

In [152]:
```
# change tuple elements ..so here we need to concert into list first then perform
```

In [153]:
```
t
```

Out[153]:
```
(5, 11, 17, 23, 29, 35, 41, 47, 53, 59)
```

In [154]:
```
t=list(t)
```

In [155]:
```
t
```

Out[155]:
```
[5, 11, 17, 23, 29, 35, 41, 47, 53, 59]
```

In [156]:
```
t.append(4)
```

In [157]:
```
t
```

Out[157]:
```
[5, 11, 17, 23, 29, 35, 41, 47, 53, 59, 4]
```

In [158]:
```
t.append([4,5,6])
```

In [159]:
```
t
```

Out[159]:
```
[5, 11, 17, 23, 29, 35, 41, 47, 53, 59, 4, [4, 5, 6]]
```

In [160]:
```
t1=tuple(range(10,20,2))
```

In [161]:
```
t1
```

Out[161]:
```
(10, 12, 14, 16, 18)
```

In [162]:
```
t.extend(t1)
t
```

Out[162]:
```
[5, 11, 17, 23, 29, 35, 41, 47, 53, 59, 4, [4, 5, 6], 10, 12, 14, 16, 18]
```

In [163]:
```
t.insert(4,5656)
```

In [164]:
```
t
```

Out[164]:
```
[5, 11, 17, 23, 5656, 29, 35, 41, 47, 53, 59, 4, [4, 5, 6], 10, 12, 14, 16, 18]
```

In [165]:
```
t=tuple(t)
```

In [166]:
```
t
```

Out[166]:
```
(5, 11, 17, 23, 5656, 29, 35, 41, 47, 53, 59, 4, [4, 5, 6], 10, 12, 14, 16, 18)
```

In [167]:
```
type(t)
```

Out[167]:
```
tuple
```

In [168]:
```
# remove tuple items
```

In [169]:
```
t=list(t)
```

In [170]:
```
t
```

Out[170]:
```
[5, 11, 17, 23, 5656, 29, 35, 41, 47, 53, 59, 4, [4, 5, 6], 10, 12, 14, 16, 18]
```

In [171]:
```
t.remove(59)
```

In [172]:
```
t
```

Out[172]:
```
[5, 11, 17, 23, 5656, 29, 35, 41, 47, 53, 4, [4, 5, 6], 10, 12, 14, 16, 18]
```

In [173]:
```
t.pop()
```

Out[173]:
```
18
```

```
In [174]: t.pop(11)
Out[174]: [4, 5, 6]

In [175]: t
Out[175]: [5, 11, 17, 23, 5656, 29, 35, 41, 47, 53, 4, 10, 12, 14, 16]

In [176]: del t[0]

In [177]: t
Out[177]: [11, 17, 23, 5656, 29, 35, 41, 47, 53, 4, 10, 12, 14, 16]

In [178]: tt=t

In [179]: tt
Out[179]: [11, 17, 23, 5656, 29, 35, 41, 47, 53, 4, 10, 12, 14, 16]

In [180]: del tt

In [181]: ##tt

In [182]: tt=t

In [183]: tt
Out[183]: [11, 17, 23, 5656, 29, 35, 41, 47, 53, 4, 10, 12, 14, 16]

In [184]: tt.clear()

In [185]: tt
Out[185]: []

In [186]: t
Out[186]: []

In [187]: t=[11, 17, 23, 5656, 29, 35, 41, 47, 53, 4, 10, 12, 14, 16]

In [188]: t=tuple(t)

In [189]: type(t)
Out[189]: tuple

In [190]: # packing and unpacking
```

In [191]:
```
f=("a","b","c","d",45,236,2.3)
```

In [192]:
```
f
```

Out[192]:
```
('a', 'b', 'c', 'd', 45, 236, 2.3)
```

In [193]:
```
type(f)
```

Out[193]:
```
tuple
```

In [194]:
```
x,y,*z=f
```

In [195]:
```
x
```

Out[195]:
```
'a'
```

In [196]:
```
y
```

Out[196]:
```
'b'
```

In [197]:
```
z
```

Out[197]:
```
['c', 'd', 45, 236, 2.3]
```

In [198]:
```
# loops
```

In [199]:
```
# loop through tuple
for i in z:
    print(i)
```

```
c
d
45
236
2.3
```

In [200]:
```
# loop through index
for i in range(len(z)):
    print(z[i])
```

```
c
d
45
236
2.3
```

In [ ]:
```

```

In [201]:
```
i=0
while i<len(z):
    print(z[i])
    i+=1
```

```
c
d
45
236
2.3
```

In [202]:
```
# join tuples
```

In [203]:
```
t3=t+tuple(z)
```

In [204]:
```
t3
```

Out[204]:
```
(11,
 17,
 23,
 5656,
 29,
 35,
 41,
 47,
 53,
 4,
 10,
 12,
 14,
 16,
 'c',
 'd',
 45,
 236,
 2.3)
```

In [205]:
```
l*2
```

Out[205]:
```
['fGHBJhjk', 'fghjFFGH', 'SFGHDJfgh', 'fGHBJhjk', 'fghjFFGH', 'SFGHDJfgh']
```

In [206]:
```
yy=t3*2
```

In [207]:
```
# count
x=yy.count(12)
```

In [208]:
```
x
```

Out[208]:
```
2
```

In [209]:
```
# postion
yy.index(12)
```

Out[209]: 11

Sets

accep all data types,unordered, unchangeble, not allow duplicates,un indexed

but we can remove and add elements

In [210]:
```
myset={1,2,3,4,5,5.6,"a","f"}
```

In [211]:
```
# automatically duplicates eliminated
myset
```

Out[211]: {1, 2, 3, 4, 5, 5.6, 'a', 'f'}

In [212]:
```
len(myset)
```

Out[212]: 8

In [213]:
```
type(myset)
```

Out[213]: set

In [214]:
```
if 61 in myset:
    print("yes")
else:
    print("no")
```

no

In [215]:
```
#constructure
sett=set(("s","d","fg","asd",4,5,6,2,3,2,2,3,6,4,5))
```

In [216]:
```
sett
```

Out[216]: {2, 3, 4, 5, 6, 'asd', 'd', 'fg', 's'}

In [217]:
```
# access elements
for x in sett:
    print(x)
```

```
2
3
4
5
6
fg
asd
d
s
```

In [218]:
```
print(2 in sett)
```

```
True
```

In [219]:
```
sett.remove(2)
```

In [220]:
```
sett
```

Out[220]:
```
{3, 4, 5, 6, 'asd', 'd', 'fg', 's'}
```

In [221]:
```
sett.discard(6)
```

In [222]:
```
sett
```

Out[222]:
```
{3, 4, 5, 'asd', 'd', 'fg', 's'}
```

In [223]:
```
sett.pop()
```

Out[223]:
```
3
```

In [224]:
```
sett
```

Out[224]:
```
{4, 5, 'asd', 'd', 'fg', 's'}
```

In [225]:
```
sett
```

Out[225]:
```
{4, 5, 'asd', 'd', 'fg', 's'}
```

In [226]:
```
sett
```

Out[226]:
```
{4, 5, 'asd', 'd', 'fg', 's'}
```

In [227]:
```
sett.clear()
```

In [228]:
```
sett
```

Out[228]:
```
set()
```

In [229]:
```
del sett
```

In [230]:
```
## sett
```

In [231]:
```
# loops
```

In [232]:
```
for x in myset:
    print(x)
```

```
1
2
3
4
5
5.6
f
a
```

In [233]:
```
# join sets
```

In [234]:
```
set1=set(range(5,96,8))
set2=set(range(10,90,9))
```

In [235]:
```
set1
```

Out[235]:
```
{5, 13, 21, 29, 37, 45, 53, 61, 69, 77, 85, 93}
```

In [236]:
```
set2
```

Out[236]:
```
{10, 19, 28, 37, 46, 55, 64, 73, 82}
```

In [237]:
```
set1.union(set2)
```

Out[237]:
```
{5, 10, 13, 19, 21, 28, 29, 37, 45, 46, 53, 55, 61, 64, 69, 73, 77, 82, 85, 93}
```

In [238]:
```
set1.update(set2)
```

In [239]:
```
set1
```

Out[239]:
```
{5, 10, 13, 19, 21, 28, 29, 37, 45, 46, 53, 55, 61, 64, 69, 73, 77, 82, 85, 93}
```

In [240]:
```
set2
```

Out[240]:
```
{10, 19, 28, 37, 46, 55, 64, 73, 82}
```

In [241]:
```
set1=set(range(5,96,8))
set2=set(range(10,90,9))
```

In [242]:
```
set1.intersection(set2)
```

Out[242]:
```
{37}
```

In [243]:
```
set1
```

Out[243]:
```
{5, 13, 21, 29, 37, 45, 53, 61, 69, 77, 85, 93}
```

In [244]:
```
set2
```

Out[244]:
```
{10, 19, 28, 37, 46, 55, 64, 73, 82}
```

In [245]:
```
set1.intersection_update(set2)
```

In [246]:
```
set1
```

Out[246]:
```
{37}
```

In [247]:
```
set2
```

Out[247]:
```
{10, 19, 28, 37, 46, 55, 64, 73, 82}
```

In [248]:
```
set1=set(range(5,96,8))
set2=set(range(10,90,9))
```

In [249]:
```
set1.symmetric_difference(set2)
```

Out[249]:
```
{5, 10, 13, 19, 21, 28, 29, 45, 46, 53, 55, 61, 64, 69, 73, 77, 82, 85, 93}
```

In [250]:
```
set1.symmetric_difference_update(set2)
```

In [251]:
```
set1
```

Out[251]:
```
{5, 10, 13, 19, 21, 28, 29, 45, 46, 53, 55, 61, 64, 69, 73, 77, 82, 85, 93}
```

In [252]:
```
set2
```

Out[252]:
```
{10, 19, 28, 37, 46, 55, 64, 73, 82}
```

In [253]:
```
#add
set1.add("hai")
```

In [254]:
```
set1
```

Out[254]:
```
{10,
 13,
 19,
 21,
 28,
 29,
 45,
 46,
 5,
 53,
 55,
 61,
 64,
 69,
 73,
 77,
 82,
 85,
 93,
 'hai'}
```

In [255]:
```
# clear
s={"fdd","rfecd",5,6,"gfdf"}
```

In [256]:
```
s
```

Out[256]:
```
{5, 6, 'fdd', 'gfdf', 'rfecd'}
```

In [257]:
```
s.clear()
```

In [258]:
```
s
```

Out[258]:
```
set()
```

In [259]:
```
del s
```

In [260]:
```
#s
```

In [261]:
```
s=set1.copy()
```

In [262]:
```
s
```

Out[262]:
```
{10,
 13,
 19,
 21,
 28,
 29,
 45,
 46,
 5,
 53,
 55,
 61,
 64,
 69,
 73,
 77,
 82,
 85,
 93,
 'hai'}
```

In [263]:
```
s=set(set1)
```

In [264]:
```
s
```

Out[264]:
```
{10,
 13,
 19,
 21,
 28,
 29,
 45,
 46,
 5,
 53,
 55,
 61,
 64,
 69,
 73,
 77,
 82,
 85,
 93,
 'hai'}
```

In [265]:
```
set1=set(range(5,96,8))
set2=set(range(10,90,9))
```

In [266]:
```
set1.difference(set2)
```

Out[266]:
```
{5, 13, 21, 29, 45, 53, 61, 69, 77, 85, 93}
```

In [267]:
```
set1
```

Out[267]:
```
{5, 13, 21, 29, 37, 45, 53, 61, 69, 77, 85, 93}
```

In [268]:
```
set1.isdisjoint(set2)
```

Out[268]:
```
False
```

In [269]:
```
set1.issubset(set2)
```

Out[269]:
```
False
```

In [270]:
```
s1={1,2,3,4,5,6}
s2={1,2,3}
```

In [271]:
```
s1.issubset(s2)
```

Out[271]:
```
False
```

In [272]:
```
s2.issubset(s1)
```

Out[272]:
```
True
```

In [273]:
```
s1.issuperset(s2)
```

Out[273]:
```
True
```

Dictionaries

accep all data types,ordered, changeble,not allow duplicates,un indexed

get(),keys(),items(),pop(),popitem(),clear(),del,
update(),values(),fromkeys(),setdefault()

In [274]:
```
dict1={"a":"chadnu","b":"shekhar","year":2022}
```

In [275]:
```
dict1
```

Out[275]:
```
{'a': 'chadnu', 'b': 'shekhar', 'year': 2022}
```

In [276]:
```
dict1["a"]
```

Out[276]:
```
'chadnu'
```

In [277]:
```
dict1["year"]
```

Out[277]:
```
2022
```

In [278]:
```
len(dict1)
```

Out[278]:
```
3
```

In [279]:
```
type(dict1)
```

Out[279]:
```
dict
```

In [280]:
```
# adding and access of elemens
dict2={"brand":"ford",
       "model":"musting",
       "year":2017,
       "colors":["a","b","c"]

}
print(dict2)
dict2["usedby"]="shekhar"
print(dict2)
```

```
{'brand': 'ford', 'model': 'musting', 'year': 2017, 'colors': ['a', 'b', 'c']}
{'brand': 'ford', 'model': 'musting', 'year': 2017, 'colors': ['a', 'b', 'c'],
'usedby': 'shekhar'}
```

In [281]:
```
dict3={"a":"gfd","b":"vfds","c":"bvfsdca","d":532,"e":352}
print(dict3)
dict3["d"]=2000
print(dict3)
```

```
{'a': 'gfd', 'b': 'vfds', 'c': 'bvfsdca', 'd': 532, 'e': 352}
{'a': 'gfd', 'b': 'vfds', 'c': 'bvfsdca', 'd': 2000, 'e': 352}
```

In [282]:
```
dict3.get("a")
```

Out[282]:
```
'gfd'
```

In [283]:
```
dict3.get("c")
```

Out[283]:
```
'bvfsdca'
```

In [284]:
```
dict3.keys()
```

Out[284]:
```
dict_keys(['a', 'b', 'c', 'd', 'e'])
```

In [285]:
```
dict3.values()
```

Out[285]:
```
dict_values(['gfd', 'vfds', 'bvfsdca', 2000, 352])
```

In [286]:
```
dict3.items()
```

Out[286]:
```
dict_items([('a', 'gfd'), ('b', 'vfds'), ('c', 'bvfsdca'), ('d', 2000), ('e', 3
52)])
```

In [287]:
```
if "a" in dict3:
    print("yes")
else:
    printint("no")
```

```
yes
```

In [288]:
```
# change vaues
```

In [289]:
```
dict3["b"]="vgbhnjmk56"
```

In [290]:
```
dict3["b"]
```

Out[290]:
```
'vgbhnjmk56'
```

In [291]:
```
dict3.update({"c":"vybuhnij65120"})
```

In [292]:
```
dict3
```

Out[292]:
```
{'a': 'gfd', 'b': 'vgbhnjmk56', 'c': 'vybuhnij65120', 'd': 2000, 'e': 352}
```

In [293]:
```
# removing elements
```

In [294]:
```
dict3.pop("a") #specified
```

Out[294]:
```
'gfd'
```

In [295]:
```
dict3.popitem() # last
```

Out[295]:
```
('e', 352)
```

In [296]:
```
del dict3["b"]
```

In [297]:
```
dict3
```

Out[297]:
```
{'c': 'vybuhnij65120', 'd': 2000}
```

In [298]:
```
dict3.clear()
```

In [299]:
```
dict3
```

Out[299]:
```
{}
```

In [300]:
```
del dict3
```

In [301]:
```
# dict3 # so dict3 deleted
```

In [302]:
```
#looping
```

In [303]:
```
for x in dict2:
    print(x)
```

```
brand
model
year
colors
usedby
```

In [304]:
```
for x in dict2:
    print(dict2[x])
```

```
ford
musting
2017
['a', 'b', 'c']
shekhar
```

In [305]:
```
for x in dict2.keys():
    print(x)
```

```
brand
model
year
colors
usedby
```

In [306]:
```
for x in dict2.values():
    print(x)
```

```
ford
musting
2017
['a', 'b', 'c']
shekhar
```

In [307]:
```
for x in dict2.items():
    print(x)
```

```
('brand', 'ford')
('model', 'musting')
('year', 2017)
('colors', ['a', 'b', 'c'])
('usedby', 'shekhar')
```

In [308]:
```
# copying
```

In [309]:
```
mydict=dict2.copy()
```

In [310]:
```
myddict=dict(dict2)
```

In [311]:
```
mydict
```

Out[311]:
```
{'brand': 'ford',
 'model': 'musting',
 'year': 2017,
 'colors': ['a', 'b', 'c'],
 'usedby': 'shekhar'}
```

In [312]:
```
myddict
```

Out[312]:
```
{'brand': 'ford',
 'model': 'musting',
 'year': 2017,
 'colors': ['a', 'b', 'c'],
 'usedby': 'shekhar'}
```

In [313]:
```
# nested dictionaries
```

In [314]:
```
my_family={"c1":{"name":"chandu","year":2012},
           "c2":{"name":"shekhar","year":2013},
           "c3":{"name":"kowshik","year":2015}}
```

In [315]:
```
my_family
```

Out[315]:
```
{'c1': {'name': 'chandu', 'year': 2012},
 'c2': {'name': 'shekhar', 'year': 2013},
 'c3': {'name': 'kowshik', 'year': 2015}}
```

In [316]:
```
c1= {'name': 'chandu', 'year': 2012}
c2= {'name': 'shekhar', 'year': 2013}
c3= {'name': 'kowshik', 'year': 2015}
my_ff={
        "chaild1":c1,
        "chaild2":c2,
        "chaild3":c3
    }
```

In [317]:
```
my_ff
```

Out[317]:
```
{'chaild1': {'name': 'chandu', 'year': 2012},
 'chaild2': {'name': 'shekhar', 'year': 2013},
 'chaild3': {'name': 'kowshik', 'year': 2015}}
```

In [318]:
```
x=("a","b","c","d")
y=(45,65,489,53)
a=dict.fromkeys(x,y)
```

In [319]:
```
a
```

Out[319]:
```
{'a': (45, 65, 489, 53),
 'b': (45, 65, 489, 53),
 'c': (45, 65, 489, 53),
 'd': (45, 65, 489, 53)}
```

In [320]:
```
a.setdefault("a")
```

Out[320]:
```
(45, 65, 489, 53)
```

strings

In [321]:
```
string="shekhar data science"
```

In [322]:
```
type(string)
```

Out[322]:
```
str
```

In [323]:
```
len(string)
```

Out[323]:
```
20
```

In [324]:
```
string[0]
```

Out[324]:
```
's'
```

In [325]:
```
string[10]
```

Out[325]:
```
't'
```

In [326]:
```
string[5:15]
```

Out[326]:
```
'ar data sc'
```

In [327]:
```
string[-1]
```

Out[327]:
```
'e'
```

In [328]:
```
string[8:15]
```

Out[328]:
```
'data sc'
```

In [329]:
```
str1="your's"
```

In [330]:
```
# str2='your's'
```

In [331]:
```
str44="""hai this is shekhar"""
```

In [332]:
```
str44
```

Out[332]:
```
'hai this is shekhar'
```

In [333]:
```
print("{0} is an float and {1} is an integer".format(5.2,10))
```

```
5.2 is an float and 10 is an integer
```

In [334]:
```
s="hai"
s1=" and bye"
```

In [335]:
```
ss=s+s1
ss
```

Out[335]:
```
'hai and bye'
```

In [336]:
```
min(s)
```

Out[336]:
```
'a'
```

In [337]:
```
max(ss)
```

Out[337]:
```
'y'
```

In [338]:
```
min(s)
```

Out[338]:
```
'a'
```

In [339]:
```
"a" in ss
```

Out[339]:
```
True
```

In [340]:
```
for letter in str44:
    print(letter)
```

```
h
a
i

t
h
i
s

i
s

s
h
e
k
h
a
r
```

In [341]:
```
s=str44
```

In [342]:
```
s
```

Out[342]: 'hai this is shekhar'

In [343]:
```
s.capitalize()
```

Out[343]: 'Hai this is shekhar'

In [344]:
```
s.center(30,"*")
```

Out[344]: '*****hai this is shekhar******'

Loops

if-else

```
a==b
a!=b
a<b
a<=b
a>b
a>=b
```

In [345]:
```
a=int(input())
b=int(input())
if a>b:
    print("a is grater")
else:
    print("b is grater")
```

```
12
23
b is grater
```

In [346]:
```
a=int(input())
b=int(input())
if a>b:
    print("yes")
```

```
21
36
```

In [347]:
```
a=int(input())
b=int(input())
print("a is grater") if a>b else print("b is grater")
```

```
12
23
b is grater
```

In [348]:
```
a=int(input())
b=int(input())
print("a>b") if a>b else print("a==b") if(a==b) else print("a<b")
```

```
52
36
a>b
```

In [349]:
```
a=int(input())
b=int(input())
c=int(input())
if a>b and a>c:
    print("a is grater")
```

```
23
63
22
```

In [350]:
```
a=int(input())
b=int(input())
c=int(input())
if a<b or a<c:
    print("a is not grater")
```

```
12
23
22
a is not grater
```

In [351]:
```
a=int(input())
b=int(input())
c=int(input())
if a>b:
    print("a is grater b")
    if a>c:
        print("a is grater than both b and c")
    else:
        print("c is grtater")
```

```
1
2
2
```

In [352]:
```
a=int(input())
b=int(input())
if a>b:
    pass
```

```
1
2
```

while

```
initialie
while condition:
    statements
    updation
```

In [353]:
```
n=int(input())
i=1
while i<n:
    print(i)
    i+=1
```

```
3
1
2
```

In [354]:
```
n=int(input())
i=1
while i<n:
    print(i)
    if i==7:
        break
    i+=1
```

```
10
1
2
3
4
5
6
7
```

In [355]:
```
n=int(input())
i=1
while i<n:
    print(i)
    if i%12==0:
        break
    i+=1
```

```
12
1
2
3
4
5
6
7
8
9
10
11
```

In [356]:
```
n=int(input())
i=1
while i<n:
    print(i)
    i+=1
    if i==10:
        continue
```

```
23
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
```

In [357]:
```
i = 1
while i < 6:
  print(i)
  i += 1
else:
  print("i is no longer less than 6")
```

```
1
2
3
4
5
i is no longer less than 6
```

for

In [358]:
```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
```

```
apple
banana
cherry
```

In [359]:
```
for x in "banana":
  print(x)
```

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

In [360]:
```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  print(x)
  if x == "banana":
    break
```

```
apple
banana
```

In [361]:
```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  if x == "banana":
    break
  print(x)
```

```
apple
```

In [362]:
```
fruits = ["apple", "banana", "cherry"]
for x in fruits:
  if x == "banana":
    continue
  print(x)
```

```
apple
cherry
```

In [363]:
```
for x in range(6):
  print(x)
```

```
0
1
2
3
4
5
```

In [364]:
```
for x in range(2, 6):
  print(x)
```

```
2
3
4
5
```

In [365]:
```
for x in range(2, 30, 3):
  print(x)
```

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

In [366]:
```
for x in range(6):
  print(x)
else:
  print("Finally finished!")
```

```
0
1
2
3
4
5
Finally finished!
```

In [367]:
```
for x in range(6):
  if x == 3: break
  print(x)
else:
  print("Finally finished!")
```

```
0
1
2
```

In [368]:
```
adj = ["red", "big", "tasty"]
fruits = ["apple", "banana", "cherry"]

for x in adj:
  for y in fruits:
    print(x, y)
```

```
red apple
red banana
red cherry
big apple
big banana
big cherry
tasty apple
tasty banana
tasty cherry
```

In [369]:
```
for x in [0, 1, 2]:
  pass
```

Functions

def

In [370]:
```
def myfunc():
    print("hai")
myfunc()
```

```
hai
```

In [371]:
```
def func(str):
    print("hai this is "+str+" @@@")
```

In [372]:
```
func("kowshik")
```

```
hai this is kowshik @@@
```

In [373]:
```
def add(a,b):
    print(a+b)
```

In [374]:
```
add(3,4)
```

```
7
```

In [375]:
```
def mul(a,b=105):
    print(a,b)
    c=a*b
    print(c)
```

In [376]:
```
mul(6)
```

```
6 105
630
```

In [377]:
```
mul(5,2)
```

```
5 2
10
```

In [378]:
```
 def addd(a,b,c):
    print(a+b+c)
```

In [379]:
```
addd(5,6,2)
```

```
13
```

In [380]:
```
addd(c=10,b=6,a=45)
```

```
61
```

In [381]:
```
def display(*marks):
    print(marks)
```

In [382]:
```
display(12,3,5,6,8,7,8,5,9,10)
```

```
(12, 3, 5, 6, 8, 7, 8, 5, 9, 10)
```

In [383]:
```
def display(a,b):
    return a,b
res=display(10,20)
print(res)
```

```
(10, 20)
```

In [384]:
```
a=100
def myf():
    print(a)
myf()
```

```
100
```

In [385]:
```
a=100
def myf():
    print(a)
myf()
print(a)
```

```
100
100
```

In [386]:
```
a=100
def myf():
    a=50
    print(a)
myf()
print(a)
```

```
50
100
```

In [387]:
```
def myf():
    m=20
    print(m)
myf()
#print(m)
```

```
20
```

In [388]:
```
def myf():
    global m
    m=20
    print(m)
myf()
print(m)
```

```
20
20
```

lambda

In [389]:
```
add=lambda a,b:a+b
```

In [390]:
```
add(5,6)
```

Out[390]: 11

In [391]:
```
l1=list(range(5,44,2))
```

In [392]:
```
l2=list(range(5,44,2))
```

In [393]:
```
len(l1)
```

Out[393]: 20

In [394]:
```
len(l2)
```

Out[394]: 20

In [395]:
```
x=lambda a:a+10
```

In [396]:
```
x(5)
```

Out[396]:
```
15
```

In [397]:
```
x=lambda a,b,c:a+b+c
x(5,6,4)
```

Out[397]:
```
15
```

In [398]:
```
def mf(n):
    return lambda a:a*n
```

In [399]:
```
mydf=mf(4)
print(mydf(2))
```

```
8
```

In [400]:
```
myt=mf(3)
print(myt(5))
```

```
15
```

In [401]:
```
mydf=mf(4)
myt=mf(3)
print(mydf(2))
print(myt(5))
```

```
8
15
```

recurtion

In [402]:
```
# factorial
def factorial(n):
    if n==0 or n==1:
        return 1
    else:
        return n*factorial(n-1)
```

In [403]:
```
factorial(5)
```

Out[403]:
```
120
```

Files

```
read--r
write--w
append--a
open()
close()
write()
writwlines()
read()
readline()
readlines()
tell()
seek()
```

In [404]:
```
f=open("shekhar.txt","w")
f.write("welcome")
f.write("good morning")
l=["hai\n","this\n","is\n","shekhar\n"]
f.writelines(l)
f.close()
```

In [405]:
```
f
```

Out[405]:
```
<_io.TextIOWrapper name='shekhar.txt' mode='w' encoding='cp1252'>
```

In [406]:
```
ff=open("kowshik.txt","w")
ff.write("dady ")
ff.write("mummy\n")
l=("hai\n","mummy\n","dady\n","how\n","are\n","you?\n")
ff.writelines(l)
ff.close()
```

In [407]:
```
ff
```

Out[407]:
```
<_io.TextIOWrapper name='kowshik.txt' mode='w' encoding='cp1252'>
```

In [408]:
```
fo=open("kowshik.txt","r")
print(fo.read(12))
fo.close()
```

```
dady mummy
h
```

In [409]:
```
f1=open("kowshik.txt","r")
print(f1.read(1))
print(f1.readline())
print(f1.readline())
print(f1.readline())
print(f1.readline())
f1.close()
```

```
d
ady mummy

hai

mummy

dady
```

In [410]:
```
f1=open("kowshik.txt","r")
print(f1.readlines())
f1.close()
```

```
['dady mummy\n', 'hai\n', 'mummy\n', 'dady\n', 'how\n', 'are\n', 'you?\n']
```

In [411]:
```
foo=open("srinaiah.txt","w")
foo.write("hai this is srinaiah")
print(foo.tell())
print(foo.seek(12))
print(foo.tell())
foo.close()
```

```
20
12
12
```

In [ ]:
```

```