Exercise NO#2.1

In [12]:

```
# A
sol=1+2+3+4+5
print(sol)

# B
sara=23
mark=19
fatima=31
avg_age=(sara+mark+fatima/3)
print(avg_age)

# C
a=73
b=403
nt=b//a
print(nt)

# D
a=73
b=403
nt=b%a
print(nt)

# E
x=2**10
print(x)

# F
mark=57
sara=54
avg_height=mark-sara
print(avg_height)

# G
p1=34.99
p2=29.95
p3=31.50
min(p1,p2,p3)
```

```
15
52.333333333333336
5
38
1024
3
```

Out[12]:

```
29.95
```

Exercise NO#2.4

In [32]:

```
s1="ant"
s2="bat"
s3="cod"

# A
sumStr=s1+s2+s3
s1Mul=s1*3
s2Mul=s2*3
s3Mul=s3*3
print(sumStr)
print(s1Mul,s2Mul,s3Mul)

# B
sumStr=s1+s1+s1+s1+s1+s1+s1+s1+s1+s1
mul=s1*10
print(sumStr)
print(mul)

# C
sumStr=s1+s2+s2+s3+s3+s3
s1Mul=s1*1
s2Mul=s2*2
s3Mul=s3*3
print(sumStr)
print(s1Mul,s2Mul,s3Mul)

# D
sumStr=s1+s2+s1+s2+s1+s2+s1+s2+s1+s2+s1+s2+s1+s2
mul=('ant''bat')*7
print(sumStr)
print(mul)

# E
sumStr=(s2*2+s3)+(s2*2+s3)+(s2*2+s3)+(s2*2+s3)+(s2*2+s3)
mul=('bat''bat''cod')*5
print(sumStr)
print(mul)
```

```
antbatcod
antantant batbatbat codcodcod
antantantantantantantantantant
antantantantantantantantantant
antbatbatcodcodcod
ant batbat codcodcod
antbatantbatantbatantbatantbatantbatantbat
antbatantbatantbatantbatantbatantbatantbat
batbatcodbatbatcodbatbatcodbatbatcodbatbatcod
batbatcodbatbatcodbatbatcodbatbatcodbatbatcod
```

Exercise NO#2.3

```
In [33]:
```

```
# A
a=3

# B
b=4

sum_mul=a*a+b*b
print(sum_mul)
```

```
25
```

Exercise NO#2.2

In [75]:

```
# A
a=2
b=2
sumab=a+b
d=sumab<a+b
print(d)


# B
b=7//3
d=b==1+1
print(d)

# C
a=3**2
b=4**2
sumab=a+b
d=sumab==25
print(d)

# D
a=2
b=4
c=6
sumab=a+b+c
d=sumab==12
print(d)

# E
a=1387
b=19
a/b==0

# F
a=31
d=31/2
s=d is "even"
print(s)

# G
a=34.99
b=29.56
c=31.50
d=min(a,b,c)<30.00
print(d)
```

```
False
True
True
True
False
True
```

Exercise NO#2.5

In [104]:

```
a=("0123456789")
# A
print(a[0])
# B
print(a[1])
# C
print(a[2])
# D
print(a[3])
# E
print(a[4])
```

```
0
1
2
3
4
```

Exercise NO#2.6

In [87]:

```
a={"bat","ball","barn","basket","badminton"}
(a)
```

Out[87]:

```
{'badminton', 'ball', 'barn', 'basket', 'bat'}
```

Exercise NO#2.7

In [103]:

```
grades=[9,7,7,10,3,9,6,6,2]

# A
a=grades.count(7)
print(a)

# B
a=grades[-1]=4
print(a)

# C
a=max(grades)
print(a)

# D
a=grades.sort()
print(a)

# E
import statistics
print(statistics.mean(grades))
```

```
2
4
10
None
6.777777777777778
```

Exercise NO#2.9

In [109]:

```
# A
A=False+False
print(type(A))

# B
A=2*3**2.0
print(type(A))

# C
A=4//2+4%2
print(type(A))

# D
A=2+3==4 or 5>=5
print(type(A))
```

```
<class 'int'>
<class 'float'>
<class 'int'>
<class 'bool'>
```

Exercise NO#2.10

In [118]:

```
# A
import math
a=3
b=6
c=a**2+b**2
d=math.sqrt(c)
print(d)

# B
if d==5:
    print("The value of hypotenuse is 5")
else:
    print("The value if hypotenuse is not 5")

# C
from math import pi
a=10
A=pi*a**2
print(A,str('sq.meter'))

# D
r=10
x=2
y=1
a=x+y==r
b=x**2+y**2==r**2
print(a)
print(b)
```

```
6.708203932499369
The value if hypotenuse is not 5
314.1592653589793 sq.meter
False
False
```

Exercise NO#2.8

In [1]:

```
# A
print('((2+3) ==4) or (a >=5)')
# B
print('(((lst[1]) * (-3)) <(-10)) ==0')
# C
print("(((lst[1]) * (-3)) <(-10)) in [0,true]")
# D
print('2 * (3**2)')
# E
print('(4/2) in [1,2,3]')
```

```
((2+3) ==4) or (a >=5)
(((lst[1]) * (-3)) <(-10)) ==0
(((lst[1]) * (-3)) <(-10)) in [0,true]
2 * (3**2)
(4/2) in [1,2,3]
```

Exercise NO#2.11

In [2]:

```
lst =[-7-6-5-4-3-2-1]
print(lst)
average_age = 9*17+24*10+21*11+27*12
x = average1 = average_age/17+24+21+27
print(average1)
a = 2**-20
print(a)
b = 4356//61
print(b)
c = 4365%61
print(c)
```

```
[-28]
127.76470588235294
9.5367431640625e-07
71
34
```

Exercise NO#2.12

In [3]:

```
a = '+'
b = '-'
print(a+b)
print(b+a+b)
print(a+b+b)
print(a+b+b+a+b+b)
print(a+b+b+a+b+b+a+b+b+a+b+b+a+b+b)
```

```
+-
-+-
+--
+--+--
+--+--+--+--+--
```

Exercise NO#2.13

In [4]:

```
s = 'abcdefghijklmnopqrstuvwxyz'
print(s[0])
print(s[2])
print(s[24])
print(s[25])
print(s[16])
```

```
a
c
y
z
q
```

Exercise NO#2.14

In [5]:

```
s = 'goodbye'
print(s[0]=='g')
print(s[6]=='g')
print(s[0]=='g' and s[1]=='a')
print(s[-2]=='x')
print(s[3]=='d')
print(s[0]==s[1])
print(s[1]+s[2]+s[3]+s[4]=='tion')
```

```
True
False
False
False
True
False
False
```

Exercise NO#2.15

In [6]:

```
a = 'anachronistically'
b = 'counterintutive'
print("number of characters : "+str(len(a)))
print("the number of characters : "+str(len(b)))

dictionary = ['misrepresentation','mis interpretation' ]
dictionary.sort()
print("\dictionary gives the word in yhis order")
print(dictionary)
word = 'floccinaucinihilipilification'
print("the number of time e repeates in is "+str(word))
word1 = 'counterrevolution'
word2 = 'resolution'
print("the number of characters are : " +str(len(word1)))
print("the number of characters arev : "+str(len(word2)))
```

```
number of characters : 17
the number of characters : 15
\dictionary gives the word in yhis order
['mis interpretation', 'misrepresentation']
the number of time e repeates in is floccinaucinihilipilification
the number of characters are : 17
the number of characters arev : 10
```

Exercise NO#2.16

In [7]:

```
a,b =6,7
c = (a+b)/2
inventory = ["paper","staples","pencils"]
first, middle, last = "John","'Fitzgerald","kennedy"
fullname = first+" "+middle+" "+last
print(fullname)
```

```
John 'Fitzgerald kennedy
```

Exercise NO#2.17

In [10]:

```
a=17-9 < 10
print(a)
a=len(inventory) > 5*len(fullname)
print(a)
a=c <= 24
print(a)
a=(6.75 < a and 6.75 > b) or (6.75 < b and 6.75 > a)
print(a)
a=len(middle) > len(first) and len(middle) < len(last)
print(a)
a=len(inventory)==0 or len(inventory) > 10
print(a)
```

```
True
False
True
True
False
False
```

Exercise NO#2.18

In [12]:

```
flowers = ['rose','bougainvillea','yucca','marigold','daylilly','liiy of the valley']
# A
print(flowers)
# B
print('potato' in flowers)
thorny=flowers[:3]
# C
print(thorny)
poisnous = flowers[-1]
# D
print(poisnous)
dangerous = [thorny] + [poisnous]
```

```
['rose', 'bougainvillea', 'yucca', 'marigold', 'daylilly', 'liiy of the va
lley']
False
['rose', 'bougainvillea', 'yucca']
liiy of the valley
```

Exercise NO#2.19

In [13]:

```
answers = ['Y','N','N','Y','N','Y','Y','Y','N','N','N']
numYes=answers.count('y')
# A
print(numYes)
numNo=answers.count('N')
# B
print(numNo)
percentYes=(numYes/len(answers))*100
# C
print(percentYes,'%')
answers.sort()
# D
print(answers)
f=answers.count('N')
# E
print("at",f,answers[f],"will occur for the first time")
```

```
0
6
0.0 %
['N', 'N', 'N', 'N', 'N', 'N', 'Y', 'Y', 'Y', 'Y', 'Y']
at 6 Y will occur for the first time
```

Exercise NO#2.20

In [16]:

```
s='pot'
print(s[::-1])
```

```
top
```

Exercise NO#2.21

In [1]:

```
s = 'Yasir'
t = 'Khan'
print(s[0]+t[0])
```

```
YK
```

Exercise NO#2.22

In [15]:

```
lst = [3,7,-2,12]
print(max(lst)-min(lst))
```

```
14
```

Exercise NO#2.23

In [1]:

```
#insert 'Apr' between 'Mar' and 'May'
monthsL=['Jan','Feb','Mar','May']
monthsT=['Jan','Feb','Mar','May']
monthsT.insert(3,"Apr")
print(monthsT)
#append string 'Jun'
monthsL=['Jan','Feb','Mar','May']
monthsT=['Jan','Feb','Mar','May']
monthsT.append("Jun")
print(monthsT)
#pop the container
monthsL=['Jan','Feb','Mar','May']
monthsT=['Jan','Feb','Mar','May']
monthsL.pop()
monthsT.pop()
print(monthsL)
print(monthsT)
#remove the second item in the container
monthsL=['Jan','Feb','Mar','May']
monthsT=['Jan','Feb','Mar','May']
del monthsT[2]
del monthsL[2]
print(monthsT)
print(monthsL)
#reverse the order of the container
monthsL=['Jan','Feb','Mar','May']
monthsT=['Jan','Feb','Mar','May']
monthsL.reverse()
monthsT.reverse()
print(monthsL)
print(monthsT)
#sort the container
monthsL=['Jan','Feb','Mar','May']
monthsT=['Jan','Feb','Mar','May']
monthsL.sort()
monthsT.sort()
print(monthsL)
print(monthsT)
```

```
['Jan', 'Feb', 'Mar', 'Apr', 'May']
['Jan', 'Feb', 'Mar', 'May', 'Jun']
['Jan', 'Feb', 'Mar']
['Jan', 'Feb', 'Mar']
['Jan', 'Feb', 'May']
['Jan', 'Feb', 'May']
['May', 'Mar', 'Feb', 'Jan']
['May', 'Mar', 'Feb', 'Jan']
['Feb', 'Jan', 'Mar', 'May']
['Feb', 'Jan', 'Mar', 'May']
```

Exercise NO#2.27

In [14]:

```
import math
hyp = float(input("Enter Length of the Ladder : "))
theta = float(input("Enter Angle of the Ladder : "))
rad = math.pi * (theta)/180
perp = hyp * math.sin(rad)
print ("The height of the Ladder is :", perp)

# A
length1=16
degrees1=75
radians1=(3.142*degrees1)/180
print(radians1)

# B
length2=20
degrees2=0
radians2=(3.142*degrees2)/180
print(radians2)

# C
length3=24
degrees3=45
radians3=(3.142*degrees3)/180
print(radians3)

# D
length4=24
degrees4=80
radians4=(3.142*degrees4)/180
print(radians4)
```

```
Enter Length of the Ladder : 19
Enter Angle of the Ladder : 30
The height of the Ladder is : 9.499999999999998
1.3091666666666666
0.0
0.7855
1.3964444444444444
```

Exercise NO#2.28

In [16]:

```
# A
list=['1','2','3','4','5']
print(list.index('3'))

# B
print(list[2])

# C
list.sort(reverse=True)
print(list)

# D
list.remove(list[0])
list.insert(4,'1')
print(list)
```

```
2
3
['5', '4', '3', '2', '1']
['4', '3', '2', '1', '1']
```

Exercise NO#2.31

In [15]:

```
lst=[2,3,4]
lst.extend([5,6])
lst
[2,3,4,5,6]
lst2=lst.copy()
lst2
[2,3,4,5,6]
lst.clear()
lst
[]
lst2
[2,3,4,5,6]
```

Out[15]:

```
[2, 3, 4, 5, 6]
```

Exercise NO#2.29

In [24]:

```
# A
a=0==(1==2)
print(a)

# B
b=2+(3==4)+5==7
print(b)

# C
c=(1<-1)==(3>4)
print(c)
```

```
True
True
True
```