Answer to Question No.01

```
def exponent(m,n):
  if n == 0:
     return 1
  x = exponent(m, n // 2)
  if n % 2 == 0:
     return x * x
  else:
     return x * x * m

print(exponent(3,2))
```

Answer to Question No.02

```
class Trace:
  def hMB(self,h):
     if (h==0):
        print("Stop: ",h)
        return 0
     elif(h==1):
        print("Stop: ",h)
        return h
     else:
        print("Continue: ",h)
        return h + self.hMB(h-1)
#Tester
t = Trace() # Object created
print("Finally ",t.hMB(5)) # hMB method got called for object t
```

Here, h=5
At first will check if condition is true or not. As, h != 0 so, if condition is false. Then will check elif condition. As, h != 1 so, elif condition is false. So else condition will be executed.

For else condition:
print("Continue: ",h)
So, (Continue: 5) will be printed
Output1 = Continue: 5
Then in return statement
5 + self.hMB(5-1)
Again hMB method is called and h = 4
So else condition will be executed
Output2 = Continue: 4
And in return statement
5 + 4 + self.hMB(4-1)
Again hMB method is called and h = 3
So else condition will be executed
Output3 = Continue: 3

And in return statement
 5 + 4 + 3 + self.hMB(3-1)
Again hMB method is called and h = 2
So else condition will be executed
Output4 = Continue: 2
And in return statement
5 + 4 + 3 + 2 + self.hMB(2-1)
Again hMB method is called and h = 1
As h=1 so, elif condition will be executed
Output5 = Stop: 1
And in return statement
5 + 4 + 3 + 2 + 1=15
So h = 15 will be returned
So, Output6 = Finally 15

```
# Final results :
Continue:  5
Continue:  4
Continue:  3
Continue:  2
Stop:  1
Finally  15
```

Answer to Question No.03

```
def hocBuilder(height):
    if height == 1:
        return 8
    else:
        return 5+hocBuilder(height-1)

print(hocBuilder(5))
```

Answer to Question No.04

```
class Surprise:
    def mystery(self,n):
        print("h(" ,n,")")
        if(n==0):
            print("value: 0")
            return 0
        else:
            print("going down")
            temp = self.mystery(n-1)+1
            print("h(",n,") --> ",temp)
            return temp
#Tester
s = Surprise()  # Object is created
```

s.mystery(4)   # mystery method is called for object s
Here parameter 4 is passed so n=4
So first of all the print statement will be executed
Output= h(4) [as n = 4]
As n = 4 so if condition will not be executed instead else condition will be executed
Then the print line will be executed
Output = going down
Then in temp again mystery method got called
Now n is (n-1)=(4-1)=3
So
Output = h(3)
As n=3 so if condition is false
So else condition will be executed
Output = going down
Then in temp again mystery method got called
Now n is (n-1)=(3-1)=2
So
Output = h(2)
As n=2 so if condition is false
So else condition will be executed
Output = going down
Then in temp again mystery method got called
Now n is (n-1)=(2-1)=1
So
Output = h(1)
As n=1 so if condition is false
So else condition will be executed
Output = going down
Then in temp again mystery method got called
Now n is (n-1)=(1-1)=0
So
Output = h(0)
As n=0 so if condition is false
So else condition will be executed
Output = value: 0
Then 0 will be returned to mystery(0). So in temp (0+1) will be stored
temp = 0+1=1
So next output is
Output = h(1) --> 1
Then 1 will be returned to myster(1)
temp = 1+1=2
So next output is
Output = h(2) --> 2
Then 1 will be returned to myster(2)
temp = 2+1=3
So next output is
Output = h(3) --> 3
Then 1 will be returned to myster(3)
temp = 3+1=4
So next output is

```
Output = h( 4 ) -->  4

# Final result :
h( 4 )
going down
h( 3 )
going down
h( 2 )
going down
h( 1 )
going down
h( 0 )
value: 0
h( 1 ) -->  1
h( 2 ) -->  2
h( 3 ) -->  3
h( 4 ) -->  4
```

Answer to Question No.05(a)

```
def recurv(n):
    if n == 1:
        print(n, end=' ')
    else:

        recurv(n-1)
        print(n, end=' ')
        #print(recurv(n-1), n)

def line(s):
    if s == 1:
        return
    else:

        line(s-1)
        if s!=2:
            print()
        recurv(s - 1)

val = 5
s = line(val+1)
```

Answer to Question No.05(b)

```
def recurv(n, h, space):
    if n == 1:
        print('  '*(h-space), end='')
        print(n, end=' ')
    else:
        recurv(n-1, h, space)
        print(n, end=' ')

def line(s, h):
    if s == 1:
        return
    else:
        line(s-1, h)
        if s!=2:
            print()
        recurv(s-1, h, s-1)

val = 5
s = line(val+1, val)
```

Answer to Question No.06

```
class FinalQ:
    def print(self,array,idx):
        if(idx<len(array)):
            print(idx+1, end='. ')
            profit = self.calcProfit(array[idx])
        #TO DO
        if idx < len(array):
            self.print(array, idx+1)

    def calcProfit(self,investment):
        #TO DO
        if investment == 25000:
            print(f'Investment: {investment}; Profit: 0.0')
        elif investment <= 100000:
            val =
((investment-25000)/100)+((investment-25000)/100)+((investment-25000)/100)+((investmen
t-25000)/100)+(((investment-25000)/100)/2)
            print(f'Investment: {investment}; Profit: {val}')
        else:
            val1
=((investment-100000)/100)+((investment-100000)/100)+((investment-100000)/100)+((inves
tment-100000)/100)+((investment-100000)/100)+((investment-100000)/100)+((investment-1
00000)/100)+((investment-100000)/100)
            val2 = (75000+75000+75000+75000+(75000/2))/100
            print(f'Investment: {investment}; Profit: {val1+val2}')



#Tester
array=[25000,100000,250000,350000]
```

```
f = FinalQ()
f.print(array,0)
#--------------------
```

Answer to Question No.07

```
def count(arr1, arr2):
    for i in arr2:
        count = 0
        for j in arr1:
            if i>=j:
                count += 1
            else:
                break
        print(count, end=' ')

a = input()
b = input()
c = input()
Arr1 = [0]*int(a[0])
Arr2 = [0]*int(a[-1])
counter = 0
for i in b:
    if i != ' ':
        Arr1[counter] = int(i)
        counter += 1
counter = 0
for i in c:
    if i != ' ':
        Arr2[counter] = int(i)
        counter += 1
count(Arr1, Arr2)
```

Answer to Question No.08