

```
In [1]: #Task_02
import random
a = [0,1,2,3]
participants = [1,2,3,4,5,6,7]
l = len(participants)
participants = participants + [None]

temp =[]
c = 0
n = 0
value = 0
rec = len(participants)-1
def remove(source,size,idx):

    for j in range(idx,len(source)):
        source[j-1]=source[j]
while True:
    value = random.choice(a)
    if value==1:

        n = int(rec/2)
        remove(participants,1,n)
        rec = rec - 1
        c= 0
        for j in participants:
            if j!=None:
                print(j,end=' ')
                c+=1

        print()
        if c==1:
            break

1 2 4 5 6 7
1 2 5 6 7
1 5 6 7
1 6 7
6 7
7
```

```
In [2]: #Task_04
class Node:
    def __init__(self,a,b=None):
        self.element = a
        self.next = b

def printDuplicate(Node_head):
    a = []
    while Node_head is not None:
        if Node_head.element not in a:
            a.append(Node_head.element)
        else:
            print(Node_head.element)
            break
        Node_head = Node_head.next

node8 =Node(6,None)
node7 =Node(10,node8)
node6 =Node(1,node7)
node5 =Node(1,node6)
node4 =Node(10,node5)
node3 =Node(10,node4)
node2 =Node(6,node3)
node1 =Node(6,node2)
head = node1
printDuplicate(head)

6
```

```
In [3]: #Task_05
class Node:
    def __init__(self,a,b=None):
        self.element = a
        self.next = b

def remove_multiple_of_five(head):
    n = head
    c=0
    a = []
    b = 0
    temp = None
    newNode= None
    while n is not None:
        if n.element%5!=0:
            a.append(n.element)
            c+=1
            n = n.next

    newNode = head
    if c==0:
        print("null")
    while newNode!=None:

        if b>=len(a):
            newNode.element = None
        else:
            newNode.element = a[b]
            newNode = newNode.next
            b+=1
    n = head
    while n!=None:
        if n.element!=None:
            print(n.element)
            n = n.next

node6 =Node(90,None)
node5 =Node(12,node6)
node4 =Node(10,node5)
node3 =Node(35,node4)
node2 =Node(6,node3)
node1 =Node(5,node2)
head = node1

remove_multiple_of_five(head)

6
12
```

```
In [4]: #Task_06
class Node:
    def __init__(self,a,b):
        self.element = a
        self.next = b

def printingSum(head,head1):
    n = head.next
    n1 = head1.next
    s = ""
    s1 = ""
    a = 0
    c = 0
    temp = None
    newNode = None
    newNodeHead = Node(None,None)
    while n is not None:
        s = s+str(n.element)
        n = n.next
    while n1 is not None:
        s1 = s1+str(n1.element)
        n1 = n1.next
    a = str(int(s)+int(s1))

    for i in a:
        c = int(i)
        if newNodeHead.next is None:
            temp = Node(c,None)
            newNodeHead.next = temp
            newNode = temp
        else:
            newNode.next = newNode = Node(c,None)
    n = newNodeHead.next
    while n!=None:
        print(n.element)
        n = n.next

node3 =Node(3,None)
node2 =Node(5,node3)
node1 =Node(4,node2)
head = Node(None,node1)

nd3 = Node(2,None)
nd2 = Node(5,nd3)
nd1 = Node(9,nd2)
head1 = Node(None,nd1)

printingSum(head,head1)

1
4
0
5
```

```
In [5]: #Task_07
class Node:
    def __init__(self,a,b):
        self.element = a
        self.next = b

def insert(head,elem):
    n = head
    c = 1
    while True:
        if n.next==head:
            n.next = Node(elem,head)
            break
        n = n.next
    n = head
    while True:
        if c>1 and n == head:
            break
        print(n.element)
        n = n.next
        c+=1

node3 =Node(3,None)
node2 =Node(2,node3)
node1 =Node(1,node2)
head = node1
node3.next = head
insert(head,4)

1
2
3
4
```

```
In [8]: #Task_08
class Node:
    def __init__(self,a,b,c):
        self.element = a
        self.next = b
        self.prev = c

def insertBefore(head,elem,newElement):
    n = head.next
    temp = None
    while n!=head:
        if n.element == elem:
            temp = Node(newElement,n,n.prev)
            n.prev.next = temp
            n = n.next
    n = head.next
    while n!=head:
        print(n.element)
        n = n.next

node4 =Node(4,None,None)
node3 =Node(3,node4,None)
node2 =Node(2,node3,None)
node1 =Node(1,node2,None)
head = Node(None,node1,None)
node1.prev = head
node2.prev = node1
node3.prev = node2
node4.next = head
insertBefore(head,3,50)

1
2
50
3
4
```

```
In [ ]: 
```