Name: Junaed Hossain ID: 20101196 Sec: 07

Ques no 1

A:

a = 96 , b = 20

25	111	52	25	0	0	45	25	5	19	101	5	26
0	1	2	3	4	5	6	7	8	9	10	11	12

Start

Size = 11

25	111	52	25	0	0	0	45	5	19	101	5	26
0	1	2	3	4	5	6	7	8	9	10	11	12

Start

Size=10

26	111	52	25	0	0	0	0	45	5	19	101	5
0	1	2	3	4	5	6	7	8	9	10	11	12

start

size=9

5	26	111	52	0	0	0	0	0	45	5	19	101
0	1	2	3	4	5	6	7	8	9	10	11	12

start

size=8

B:

5	26	111	52	0	0	0	0	0	45	5	19	101
0	1	2	3	4	5	6	7	8	9	10	11	12

start

Right shifting the element from position 5 to the last:

5	0	26	111	52	0	0	0	0	45	5	19	101
0	1	2	3	4	5	6	7	8	9	10	11	12

adding element (b%67=20) in position 5

5	20	26	111	52	0	0	0	0	45	5	19	101
0	1	2	3	4	5	6	7	8	9	10	11	12

C:

5	20	26	111	52	0	0	0	0	45	5	19	101
0	1	2	3	4	5	6	7	8	9	10	11	12

Inserting studentId%13 = (96%13=5) at position 8.

5	20	26	111	0	52	0	0	5	45	5	19	101
0	1	2	3	4	5	6	7	8	9	10	11	12

D:

5	20	26	111	0	52	0	0	5	45	5	19	101
0	1	2	3	4	5	6	7	8	9	10	11	12

start

Inserting birthyear%61 (2000%61=48) at position 3.

101	5	20	26	111	0	52	0	5	45	5	19	48
0	1	2	3	4	5	6	7	8	9	10	11	12

start

E:

101	5	20	26	111	0	52	0	5	45	5	19	48
0	1	2	3	4	5	6	7	8	9	10	11	12

start

Removing 5 by leftshifting

101	20	26	111	0	52	0	0	5	45	5	19	48
0	1	2	3	4	5	6	7	8	9	10	11	12

start

20	26	111	0	52	0	0	0	5	45	19	48	101
0	1	2	3	4	5	6	7	8	9	10	11	12

Start

F:

20	26	111	0	52	0	0	0	5	45	19	48	101
0	1	2	3	4	5	6	7	8	9	10	11	12

Start

Removing 52 by left-shifting.

20	26	111	0	0	0	0	0	5	45	19	48	101
0	1	2	3	4	5	6	7	8	9	10	11	12

end Start

G:

20	26	111	0	0	0	0	0	5	45	19	48	101
0	1	2	3	4	5	6	7	8	9	10	11	12

Right rotating 3 times

101	20	26	111	0	0	0	0	0	5	45	19	48
0	1	2	3	4	5	6	7	8	9	10	11	12

start

48	101	20	26	111	0	0	0	0	0	5	45	19
0	1	2	3	4	5	6	7	8	9	10	11	12

start

19	48	101	20	26	111	0	0	0	0	0	5	45
0	1	2	3	4	5	6	7	8	9	10	11	12

End start

H:

19	48	101	20	26	111	0	0	0	0	0	5	45
0	1	2	3	4	5	6	7	8	9	10	11	12

End start

Left rotating 4 times

48	101	20	26	111	0	0	0	0	0	5	45	19
0	1	2	3	4	5	6	7	8	9	10	11	12

start

101	20	26	111	0	0	0	0	0	5	45	19	48
0	1	2	3	4	5	6	7	8	9	10	11	12

20	26	111	0	0	0	0	0	5	45	19	48	101
0	1	2	3	4	5	6	7	8	9	10	11	12

26	111	0	0	0	0	0	5	45	19	48	101	20
0	1	2	3	4	5	6	7	8	9	10	11	12

End Start

Ques no 2

```
import random as rd
lists=["A","B","C","D","E","F","G"]
num=1
output_list = []
while len(lists)!=1:
    for item in range(len(lists) - num, len(lists)):
        output_list.append(lists[item])
    for item in range(0, len(lists) - num):
        output_list.append(lists[item])
    lists=output_list
    music=rd.randint(0,3)
    if music==1:
        lists.pop(len(lists)//2)
        if len(lists)==1:
            print(lists)
            print("And the Winner is : ",lists[0])
        else:
            print(lists)
    output_list=[]
```

Ques no 3

Node		10	20	30	40	50
Data	J	U	N	A	E	D

Next Node	10	20	30	40	50	None

The direction is from J to D:

Head	Next node				
J	U	N	A	E	D

New head-1

Head| new node	Next node				
J	U	N	A	E	D

Head | new head	Next node				
J	U	N	A	E	D

The direction is from U to D:

Head	New head	Next node			
J	U	N	A	E	D

The direction is from N to D:

		New head	Next node		
J	U	N	A	E	D

The direction is from A to D:

			New head	Next node	
J	U	N	A	E	D

The direction is from E to D:

				New head	Next node
J	U	N	A	E	D

					New head| Next node: None
J	U	N	A	E	D

The direction is now reverse from the first time. So, the direction is now from D to J:

J	U	N	A	E	D

So, the linked list will be:

D	E	A	N	U	J

B:

Inserting P in position 1:

Head						
P	**D**	**E**	**A**	**N**	**U**	**J**

C:

Inserting A in position 2:

Head							
P	**D**	**A**	**E**	**A**	**N**	**U**	**J**

D:

Left rotate 1 time:

Head							
D	**A**	**E**	**A**	**N**	**U**	**J**	**P**

LEFT ROTATE 2 TIME:

HEAD

A	E	A	N	U	J	P	D

LEFT ROTATE 3 TIME:

HEAD

E	A	N	U	J	P	D	A

LEFT ROTATE 4 TIME:

HEAD

A	N	U	J	P	D	A	E

E: Delete the second element of the list

HEAD	head.next						
A	N	U	J	P	D	A	E

Self.head.next.element=None

A	None	U	J	P	D	A	E

F: Insert ‘G’ in the last position

Node.next=newnode(G,None)

A		U	J	P	D	A	E	G

G: Right rotate the list 3 times

A	E	G	A		U	J	P	D

H: Sort the list in alphabetical order

A	A	E	G		U	J	P	D

A	A	D	E	G		U	J	P

A	A	D	E	G		J	U	P

A	A	D	E	G		J	P	U

A	A	D	E	G	J	P	U	

Ques no 4

```
def printduplicate(self):
    n=self.head
    count=0
    while n!=None:
       m=n.element
       b=n.next

       while b!=None:
          if m==b.element:
             count+=m
             break
          else:
             b=b.next
```

```
            if count>1:
                break
            else:
                n=n.next
        print(count)
```

Ques no 5

```
def remove_multiple_of_five(self):
        n = self.head
        while(n != None and (n.element % 5) == 0):
            self.head = n.next
            del(n)
            n = self.head
        while(n != None):
            while(n != None and (n.element % 5) != 0):
                prev = n
                n = n.next
            if(n == None):
                return
            prev.next = n.next
            del(n)
            n = prev.next
```

ques no 6

```
class Node:
    def __init__(self, element, next):
        self.element = element
        self.next = None
n = Node(4,None)
```

```
n1 = Node(5,None)
n2 = Node(3,None)
#the dummy head
head1 = Node(None,None)
head1.next = n
n.next = n1
n1.next = n2


n = Node(4,None)
n1 = Node(5,None)
n2 = Node(3,None)
#the dummy head
head2 = Node(None,None)
head2.next = n
n.next = n1
n1.next = n2
#=======method==============================
def sumoftwo(head1,head2):
    p=head1.next
    q=head2.next
    #summ=0
    count=-1
    while p!=None:
        count+=1
        p=p.next
    div=10**count
    summ=0
    p=head1.next
```

```
while p!=None:
    summ+=p.element*div
    div=div//10
    p=p.next
print(summ)
count=-1
while q!=None:
    count+=1
    q=q.next
dev=10**count
q=head2.next
while q!=None:
    summ+=q.element*dev
    dev=dev//10
    q=q.next
print(summ)
sumHead=Node(None,None)

tail=None
for i in str(summ):
    new=Node(int(i),None)
    if (sumHead.next==None):
        sumHead.next=new
        tail=new
    else:
        tail.next=new
        tail=new
w=sumHead.next
while w!= None:
```

```
        print(w.element,end="->")
        w = w.next


sumoftwo(head1,head2)
```

Ques no 7

```
def insertion(self,data):
    if self.head==None:
        self.head=Node()
        self.head.next=self.head
    else:
        new=Node(data)
        n=self.head
        while n.next is not self.head:
            n=n.next
        n.next=new
        new.next=self.head
```

Ques no 8

```
  def insert_eg(self,newElement,index):
    n=self.head.next
    while n is not self.head:
        if (n.ele==index):
            break
        else:
            n=n.next
    new=Node(None,newElement,None)
    new.prev=n.prev
    new.next=n
```

n.prev.next=new

n.prev=new