StudentId = 20101238 a= 38 b= 20 BirthYear = 2000

A.

Here,  $start_index = 6$ 

									_				
Value	26	25	53	52	0	0	45	45	25	5	19	43	5
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
					-	ţ	-				-	-	
		_											
Value	26	25	53	52	0	0	0	45	25	5	19	43	5
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						1							
Value	5	26	53	52	0	0	0	45	45	25	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						<b>†</b>						<u> </u>	
											_		
Value	5	26	53	52	0	0	0	0	45	45	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						1							
Value	5	26	53	52	0	0	0	0	0	45	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12

B.

Value	5	26	53	52	0	0	0	0	0	45	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12

Start index

Here the start index of this array is 9.

b%67=20%67=20

Position 5 means index=1.

	-					-				-	-	-	
Value	5	26	53	52	52	0	0	0	0	45	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
	•		•			1			•				
Value	5	26	53	53	52	0	0	0	0	45	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						1							
Value	5	26	26	53	52	0	0	0	0	45	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
	•					1							
Value	5	20% 67= 20	26	53	52	0	0	0	0	45	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
		•			-	1		-	-				
Value	5	20	26	53	52	0	0	0	0	45	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12

C.

Start index = 9

StudentId%13=20101238%13=1

Position 8 means index = 4

Value	5	20	26	53	52	52	0	0	0	45	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
					-	+	-		·		-	·	-
Value	5	20	26	53	1	52	0	0	0	45	5	19	43

D.

Start index = 9

BirthYear = 2000

BirthYear%61=2000%61=48

Position 3 means index=12

Value	5	20	26	53	1	52	52	0	0	45	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						1							
Value	5	20	26	53	1	1	52	0	0	45	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						1							
Value	5	20	26	53	53	1	52	0	0	45	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						1							
Value	5	20	26	26	53	1	52	0	0	45	5	19	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12

ŧ

		<u> </u>	<u> </u>			1	<u> </u>	<u> </u>	<u> </u>				
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Value	43	5	20	26	53	1	52	0	0	45	5	19	48
						1							
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Value	43	5	20	26	53	1	52	0	0	45	5	19	43
						1							
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Value	5	5	20	26	53	1	52	0	0	45	5	19	43
						1							
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Value	5	20	20	26	53	1	52	0	0	45	5	19	43

E.

Value	43	5	20	26	53	1	52	0	0	45	5	19	48
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						1							
Value	43	5	20	26	53	1	52	0	0	45	19	19	48
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						+							
Value	43	5	20	26	53	1	52	0	0	45	19	48	48
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
	,					1			•				

Similarly,

Value	5	20	26	53	1	52	52	0	0	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						1							
Value	5	20	26	53	1	52	0	0	0	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						ţ							
Value	20	20	26	53	1	52	0	0	0	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
		•	•			1	•						
Value	20	26	26	53	1	52	0	0	0	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						1							
Value	20	26	53	1	52	52	0	0	0	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
		•	•			1						•	
Value	20	26	53	1	52	0	0	0	0	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
	•				•	+						•	•
F													

F.

Value	20	26	53	1	0	0	0	0	0	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12

G.

Value	20	26	53	1	0	0	0	0	0	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12

1

<b>T</b> 7 1													
Value	20	26	53	1	1	0	0	0	0	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						1							
Value	20	26	53	53	1	0	0	0	0	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						1							
Value	20	26	26	53	1	0	0	0	0	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Similarly	<b>у</b> ,					1							
Value	43	20	26	53	1	0	0	0	0	45	45	19	48
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Index 9	value = (	)				<b>†</b>							
Index 9 v	value = (	20	26	53	1	0	0	0	0	0	45	19	48
<u> </u>	T	1	26	53	1 4	Γ	0 6	0 7	0 8	0 9	45	19 11	48
Value	43	20				0							
Value Index	43	20				0							
Value Index Time = 2	43 0	20	2	3	4	5	6	7	8	9	10	11	12
Value Index Time = 2	43 0	20 1 20	26	53	1	5	6	7	8	9	10	11	12
Value Index Time = 2	43 0	20 1 20	26	53	1	0 5	6	7	8	9	10	11	12
Value Index Time = 2 Value Index	43 0 43 0	20 1 20 1	26 2	3 53 3	1 4	0 5 1 5	6 0 6	7 0 7	0 8	9 0 9	10 45 10	11 19 11	12 48 12
Value Index  Time = 2  Value Index  Value	43 0 43 0 43 0	20 1 20 1	2 26 2 26	3 53 3	1 4	0 5 1 5	6 0 6	7 0 7	8 0 8	9 0 9	10 45 10	11 19 11	12 48 12 48

Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Setting in	dex 10 y	value =	0			<b>†</b>							
Value	48	43	20	26	53	1	0	0	0	0	0	45	19
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
time=3						+							
Value	48	43	20	26	53	1	0	0	0	0	0	45	19
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						ţ							
Value	48	43	20	26	53	1	1	0	0	0	0	45	19
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						ţ							_
Value	48	43	20	26	53	53	1	0	0	0	0	45	19
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Similarly,						ţ							
Value	19	48	43	20	26	53	1	0	0	0	0	45	45
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Setting in	dex 11 v	value =	0			ţ							
Value	19	48	43	20	26	53	1	0	0	0	0	0	45
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
H. Time =1						1							
Value	19	48	43	20	26	53	1	0	0	0	0	0	45
Index	0	1	2	3	4	5	6	7	8	9	10	11	12

Value	19	48	43	20	26	53	1	0	0	0	0	45	45
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
				•		1							
Value	19	48	43	20	26	53	1	0	0	0	0	45	19
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Similarly,													
Value	48	43	20	26	53	1	1	0	0	0	0	45	19
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Setting index 6 value = 0													
Value	48	43	20	26	53	1	0	0	0	0	0	45	19
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
time=2													
Value	48	43	20	26	53	1	0	0	0	0	45	45	19
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						<u>†</u>							
						•							
Value	48	43	20	26	53	1	0	0	0	0	45	19	19
Value Index	48	43	20	26	53		0	0 7	0	9	45	19 11	19 12
	0					1							
Index	0					1 5							
Index Similarly	0	1	2	3	4	1 5	6	7	8	9	10	11	12
Index Similarly Value	43	20	26 2	53	1	1 5 •••••••••••••••••••••••••••••••••••	6	7	8	9	10	11	12
Index Similarly Value Index	43	20	26 2	53	1	1 5 1 5	6	7	8	9	10	11	12

time=3

Value	43	20	26	53	1	0	0	0	0	45	45	19	48
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						†							
Value	43	20	26	53	1	0	0	0	0	45	19	19	48
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Similarly	,					ţ							
Value	20	26	53	1	1	0	0	0	0	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Setting ir	ndex 4 v	alue =	0,			1			•				
Value	20	26	53	1	0	0	0	0	0	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
time = 4													
Value	20	26	53	1	0	0	0	0	45	45	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
						1							
Value	20	26	53	1	0	0	0	0	45	19	19	48	43
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
Similarly	,					1							
Value	26	53	1	1	0	0	0	0	45	19	48	43	20
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
<b>+</b>													
Setting in Value	26	53	0,	0	0	0	0	0	45	19	48	43	20
Index	0	1	2	3	4	5	6	7	8	9	10	11	12
-: •	_										_		

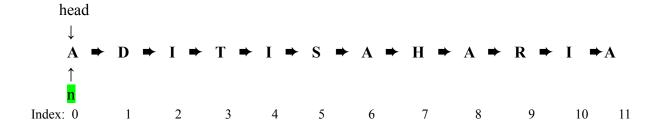


new = None

A.

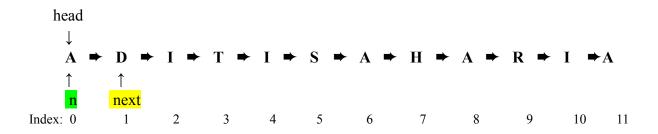
# **Step-01:**

n=head

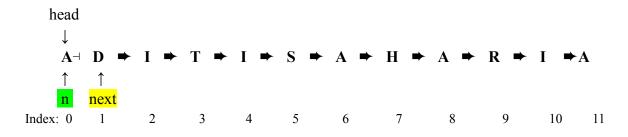


#### **Step-02:**

next = n.next

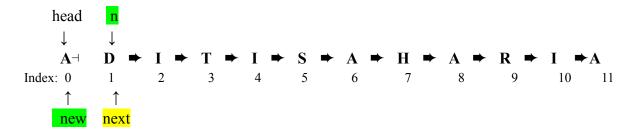


n.next = new



new = n

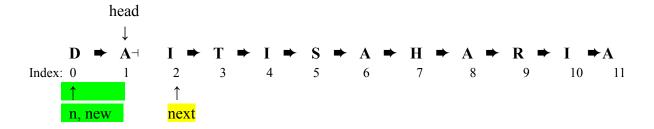
n = next



Again repeating the steps,

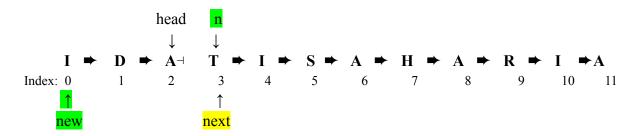
As, n.next = new so,

As, new = n so,

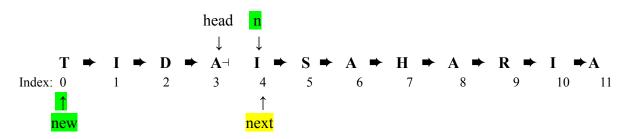


As n=next so,

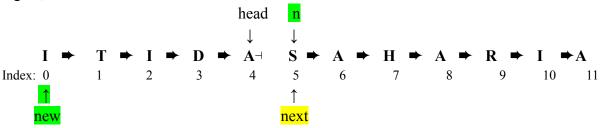
Again,



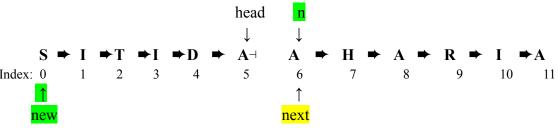
Again,



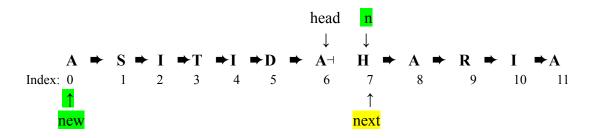
Again,



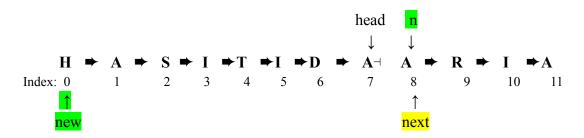
Again,



Again,



Again,



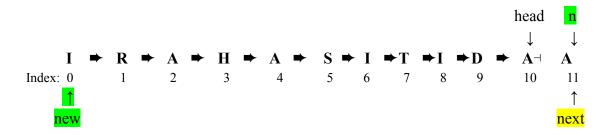
Again,

head 
$$n$$

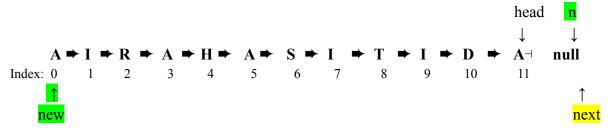
$$A \rightarrow H \rightarrow A \rightarrow S \rightarrow I \rightarrow T \rightarrow I \rightarrow D \rightarrow A \rightarrow R \rightarrow I \rightarrow A$$
Index: 0 1 2 3 4 5 6 7 8 9 10 11
$$\uparrow \qquad \qquad \uparrow \qquad \qquad \downarrow \qquad$$

Again,

Again,



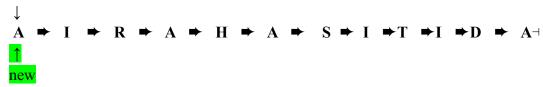
Again,

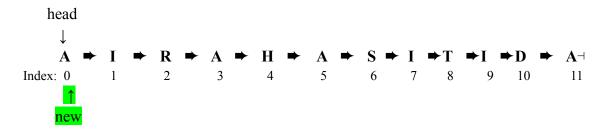


Last step:

head = new

head





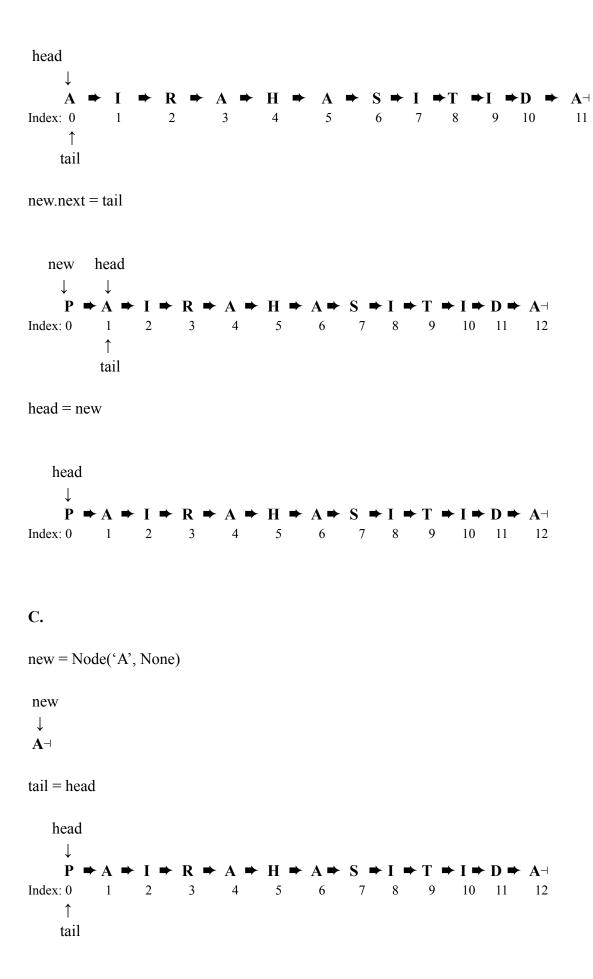
B.



Creating new Node for P, new = Node('P',None) tail = head

new

↓ P⊣

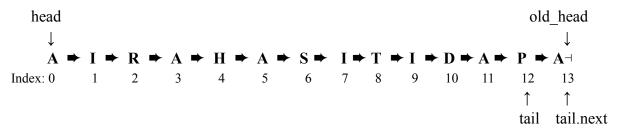


```
position = 2
for i in range(position-1):
   tail = tail.next
After the loop ends
    head
      \downarrow
                    2 3 4
                                            5 6
                                                          7
             \uparrow
            tail
pre = tail.next
tail.next = new
new.next = pre
    head
      \downarrow
                            3 4 5 6 7
Index: 0
            tail
                   pre
    head
      \downarrow
      P \Rightarrow A \Rightarrow A \dashv
Index: 0
            1
           tail
I \Rightarrow R \Rightarrow A \Rightarrow H \Rightarrow A \Rightarrow S \Rightarrow I \Rightarrow T \Rightarrow I \Rightarrow D \Rightarrow A \dashv
↑
pre
    head
         1
                                  4 5 6 7
                                                                   8
                                                                        9 10 11 12 13
                            3
            tail
                           pre
```

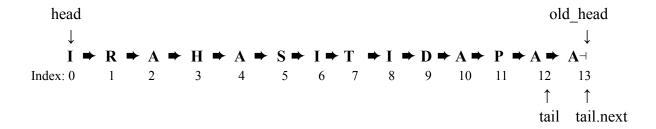
```
D.
```

for i in range(4): #loop will run 4 times Time 1: old head = headhead = head.nexttail = headold head head  $\downarrow$ 6 7 8 3 4 5 tail old\_head.next = None old head head Index: 0 3 5 6 7 tail while tail.next is not None: tail = tail.next After the end of the loop tail will be index 12 old head head 3 6 1 tail tail.next = old headhead old head 2 3 4 5 6 7 8 10 11  $\uparrow$ tail tail.next

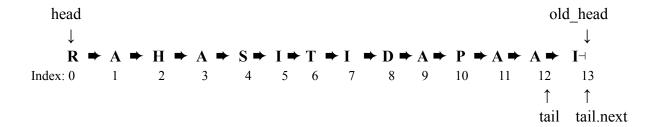
Time 2:



#### Time 3:



#### Time 4:

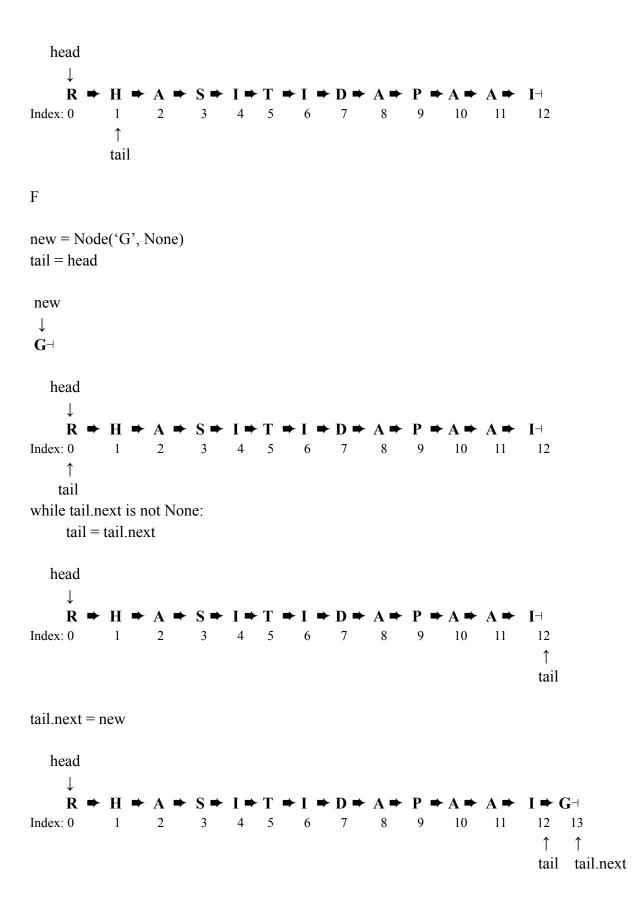


#### E.

Second elements index is 1 tail = head.next.next

head  $\downarrow \\ \mathbf{R} \Rightarrow \mathbf{A} \Rightarrow \mathbf{H} \Rightarrow \mathbf{A} \Rightarrow \mathbf{S} \Rightarrow \mathbf{I} \Rightarrow \mathbf{T} \Rightarrow \mathbf{I} \Rightarrow \mathbf{D} \Rightarrow \mathbf{A} \Rightarrow \mathbf{P} \Rightarrow \mathbf{A} \Rightarrow \mathbf{A} \Rightarrow \mathbf{I} \vdash$ Index: 0 1 2 3 4 5 6 7 8 9 10 11 12 13  $\uparrow \\ \text{tail}$ 

del = head.next del.next = None del.element = None head.next = tail



```
G.
for i in range(3):
   #loop will run 3 times
Time 1:
tail = head
new = None
   head
       \downarrow
       R \Rightarrow H \Rightarrow A \Rightarrow S \Rightarrow I \Rightarrow T \Rightarrow I \Rightarrow D \Rightarrow A \Rightarrow P \Rightarrow A \Rightarrow A \Rightarrow I \Rightarrow G \dashv
Index: 0 1 2 3 4 5 6 7 8 9 10 11
                                                                                                      12 13
       \uparrow
     tail
while tail.next is not None:
      new = tail
       tail = tail.next
   head
       R \Rightarrow H \Rightarrow A \Rightarrow S \Rightarrow I \Rightarrow T \Rightarrow I \Rightarrow D \Rightarrow A \Rightarrow P \Rightarrow A \Rightarrow A \Rightarrow I \Rightarrow G \dashv
                       2 3 4 5 6 7 8 9 10 11
Index: 0
                                                                                                      12 13
                                                                                                         \uparrow
                                                                                                               1
                                                                                                       new tail
new.next = None
   head
       R \Rightarrow H \Rightarrow A \Rightarrow S \Rightarrow I \Rightarrow T \Rightarrow I \Rightarrow D \Rightarrow A \Rightarrow P \Rightarrow A \Rightarrow A \Rightarrow I \dashv G \dashv
                                3 4 5 6 7 8 9 10 11
                                                                                                         ↑
                                                                                                       new tail
tail.next = head
      tail head
       G \Rightarrow R \Rightarrow H \Rightarrow A \Rightarrow S \Rightarrow I \Rightarrow T \Rightarrow I \Rightarrow D \Rightarrow A \Rightarrow P \Rightarrow A \Rightarrow A \Rightarrow I \dashv
                        2 3 4 5 6 7 8 9 10 11 12
Index: 0
head = tail
     head
       G \Rightarrow R \Rightarrow H \Rightarrow A \Rightarrow S \Rightarrow I \Rightarrow T \Rightarrow I \Rightarrow D \Rightarrow A \Rightarrow P \Rightarrow A \Rightarrow A \Rightarrow I \dashv
Index: 0 1 2 3 4 5 6 7 8 9 10 11 12 13
```

```
Time 2:
Repeating the same steps
      head
        \downarrow
        I \Rightarrow G \Rightarrow R \Rightarrow H \Rightarrow A \Rightarrow S \Rightarrow I \Rightarrow T \Rightarrow D \Rightarrow A \Rightarrow P \Rightarrow A \Rightarrow A \dashv
                                             4
                                                      5 6 7 8 9 10 11 12 13
Index: 0
Time 3:
Repeating the same steps
      head
A \Rightarrow I \Rightarrow G \Rightarrow R \Rightarrow H \Rightarrow A \Rightarrow S \Rightarrow I \Rightarrow T \Rightarrow I \Rightarrow D \Rightarrow A \Rightarrow P \Rightarrow A \dashv
Index: 0 1 2 3 4 5 6 7 8 9 10 11 12 13
Н.
new = None
count = 0
tail = head
while tail is not None:
       count = count+1
       tail = tail.next
count = count+1
for i in range(count):
     #loop will run 13 times
inside the loop
```

```
def printDuplicate(head):
    tail = head
    while tail is not None:
        n = tail.next
        while n is not None:
        if n.element is tail.element:
            print(n.element)
        return
```

```
else:

n = n.next

tail = tail.next
```

```
def remove_multiple_of_five(head):
 n = head
 while n is not None:
    tail = head
    new = None
    while tail is not None and tail.element%5!=0:
      new = tail
      tail = tail.next
    if new is None:
      head = tail.next
      n = head
    elif tail is None:
      new.next = None
    else:
      new.next = tail.next
    n = n.next
 if n is None and head.element%5==0:
    head = None
 return head
```

#### **Answer to Question No.6**

```
def sum(self):
    a = self.first_head.next
    b = self.scnd_head.next
    count1 = 0
    count2 = 0
    while a is not None:
        count1 += 1
        a = a.next
    a = self.first_head.next
    while b is not None:
```

```
count2 += 1
b = b.next

b = self.scnd_head.next

r1 = 0

while a is not None:

r1 = r1 + a.element*(10**(count1-1))

count1 = count1-1
a = a.next

r2 = 0

while b is not None:

r2 = r2 + b.element*(10**(count2-1))

count2 = count2-1
b = b.next

add = r1+r2

print(add)
```

```
def insert(head, newElement, index=None):
  inserted_value = Node(newElement, None)
  copy_tail = head
  for i in range(index - 1):
     copy_tail = copy_tail.next
  pre = copy_tail
  inserted_value.next = pre.next
  pre.next = inserted_value
```

#### **Answer to Question No.8**

```
def insert(head, elem, newElement):
  box = Node(newElement, None, None)
  tail = head.next
  new = None
  while tail.val is not elem and tail is not None:
     new = tail
     tail = tail.next
  pre = new
  box.next = pre.next
  pre.next.prev = box
  box.prev = pre
  pre.next = box
```