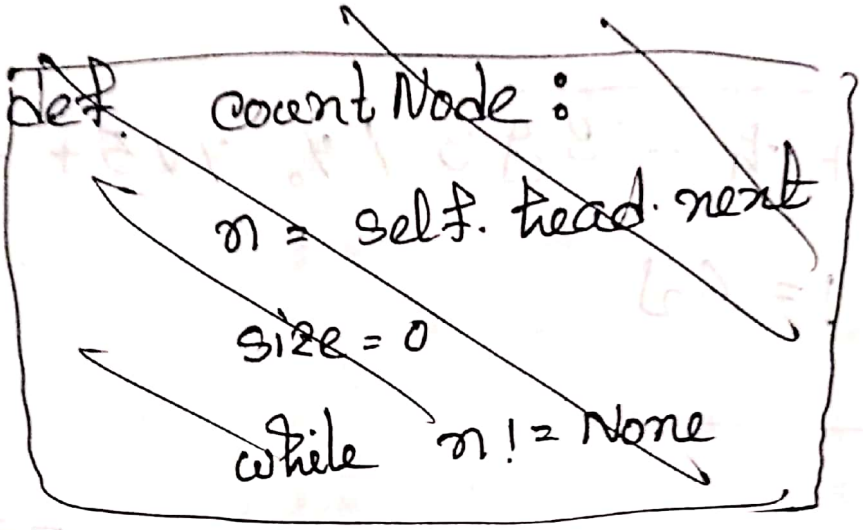
Answer to the Question no: 1

```
def checkInterval(cir_arr, size, start, int
    index = start
    i = 0
    while i< size:
        if index != start:
            if cir_arr[index] > 0:
                cir_arr[index] = -1 * cir_a
                [in
        index = (index+1) % len(cir_arr) +
                                    interv
        i += 2
    return cir_arr
cir_arr = [0, 0, -2, 4, 3, 3, 5, -6, 7, 10, 0, 0]
interval = 2
size = 8
start = 2
print(checkInterval(cir_arr, size, start, inter
```

Answer to the Question no: 2



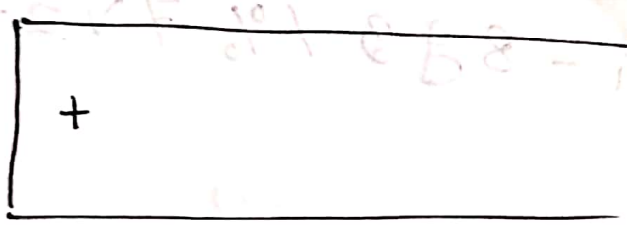

```
class Node:
    def __init__(self):
        self.data = None
        self.next = None
        self.prev = None

def countNode:
    n = self.head.next
    size = 0
    while n != None:
        size += 1
        n = n.next
    return size
```

```
def updateList (head, number)
    n = head.next
    while n != None
        if (n.data % countNode) % 2 == 0:
            temp = n.next
            n.prev.next = n.next
            temp.prev = n.prev
            n = temp
        else:
            n.data = number.
            n = n.next.
    return head.
```

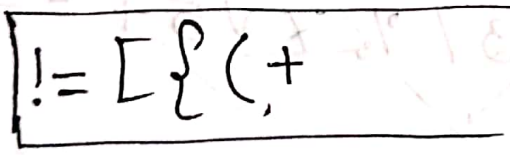
Answer to the Question no: 3

Given

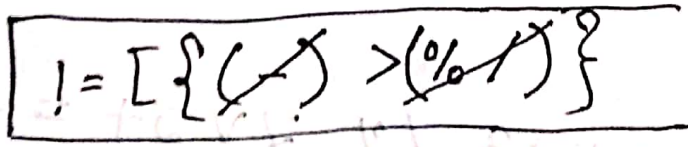
```
2+ t != [{(4+1-h)>(8% 9/3)}
        88 (7 == v+5)] || (6+ y <= k)
```



```
2 t •
```



```
!= [{(+
2t+ 9 1
```



```
!= [{(-) >(%/)}
2t +41 + h - 893/% >
```

||

!= [SS (==+)

2t + 41 + h - 8g3 /% > 7 V5 + ==

<=

!= [SS]

2t + 41 + h - 8g3 /% 7 V5 + == S

~~!= [|| (<=~~

~~2t + 41 + h - 8g3 /% 7 V5 + == SS~~

~~6 y + K~~

||

!=

2t + 41 + h - 8g3 /% 7 V5 + == SS

<=

| 11 + |

2t + 41 + h - 8g3 / % 7V5+ == 88

!= 6y

| 11 <= |

2t + 41 + h - 8g3 / % 7V5+ == 88

!= 6y+

| 11 |

2t + 41 + h - 8g3 / % 7V5+ == 88

!= 6y+ <=

| |

2t + 41 + h - 8g3 / % 7V5+ == 88

!= 6y+ <= 11