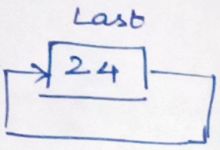
1.

DSA ASSIGNMENT

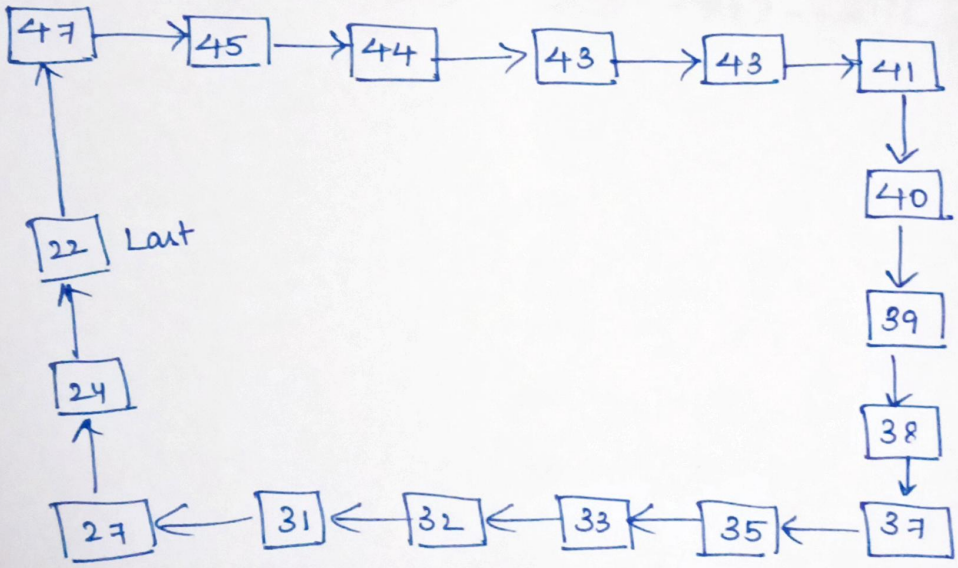
SELVAKUMAR G (22MAI1004)

1) Inserting the 3 list into a circular linked list.

Insert 24



Insert all other elements and sort them



Only 15 elements are allowed in the merge list so 22 & 24 will be deleted.

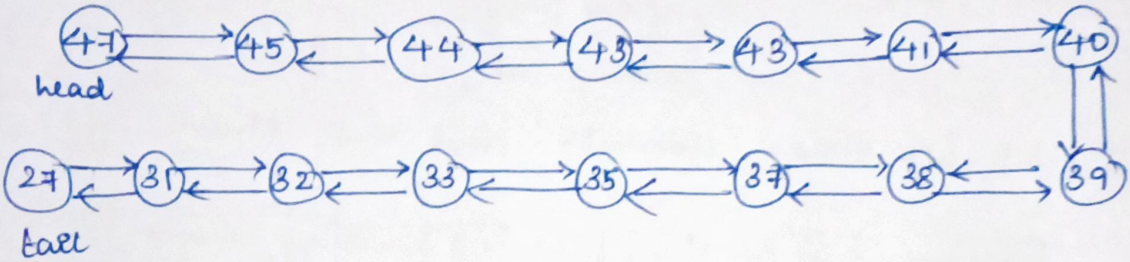
Need to traverse from the last -> next and increase the count to 15 after coming elements are removed.

Time Complexity

Insertion : $O(n^2)$

Deletion : $O(n)$

Doubly linked list



```c
#include<stdio.h>
#include<stdlib.h>

struct Node
{
    int value;
    struct Node *link;
};
struct Node *last;

void Insert(int value)
{
    struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
    newNode->value = value;

    if(last == NULL)
    {
        last = newNode;
        last->link = last;
    }
    else
    {
        struct Node *head = last->link;

        if(head->value < value)
        {
            newNode->link = head;
            last->link = newNode;
        }
        else if(last->value > value)
        {
            newNode->link = head;
            last->link = newNode;
            last = newNode;
        }
        else
        {
            struct Node *currentNode = head;
            struct Node *prevNode = last;
```

```
            last = newNode;
        }
        else
        {
            struct Node *currentNode = head;
            struct Node *prevNode = last;

            do
            {
                if(value > currentNode->value && value < prevNode->value)
                {
                    prevNode->link = newNode;
                    newNode->link = currentNode;
                    break;
                }

                prevNode = currentNode;
                currentNode = currentNode->link;
            }while(currentNode != head);
        }
    }
}

void Delete(int vacancy)
{
    struct Node *node = last->link;
    int count = 1;

while(count < vacancy)
{
    node = node->link;
    count++;
}
node->link = last->link;
last = node;
}
```

2. Infix to Postfix Expression : $(A + B) * C / D - E$

Symbol	Stack	Output
(	(	
A		A
+	(+	A
B	(+	A
)		AB+
*	*	AB+
C	*	AB+C
/	/	AB+C*
D	/	AB+C*D/
-	-	AB+C*D/
E	-	AB+C*D/E
		AB+C*D/E-

Postfix Expression : AB+C*D/E-

Evaluate : A = 3, B = 7, C = 4, D = 3 and E = 1 we get

3 + 7 = 10 (AB+)

~~10 * 4 = 40 (AB+C*)~~

10 * 4 = 40 (AB+C*)

40 / 3 = 13 (AB+C*D/)

13 - 1 = 12 (AB+C*D/E-)

Algorithm

1) Initialize an empty stack and an empty string
2) For each character in the Infix expression
   - If the character is an operand, add it to the output string.
   - If the character is an operator
     * While the stack is not empty and the top of the stack is an operator with greater or equal precedence to the current operator
   - Push the current operator onto the stack
   - If the character is an open parantheses, push it onto the stack
   - If the character is a close parentheses
     * While the stack is not empty, and the top of the stack is not an open paranthesis
     * Pop the top operator from the stack and add it to the output string
     * Pop the open paranthesis from the stack
   - While the stack is not empty
     * Pop the top operator from the stack and add it to the output string.
3) The output string should now contain the postfix expression.