**DSA LAB**

**Lab Assignment number 12**

**Name:** Aamir Ansari **Batch:** A **Roll no:** 01

**Aim:** To implement Circular Doubly linked lists

**THEORY:**

**Circular Doubly Linked List:**

Circular Doubly Linked List is a linked list in which two consecutive elements are linked or connected by previous and next pointer and the last node points to first node by next pointer and also the first node points to last node by previous pointer.

**ALGORITHM:**

1. INSERT

a)At a position:

Step 1: [INITIALIZE] temp

Step 2: IF POSITION == 1

[INITIALIZE] ptr

SET ptr->data = data

IF START == NULL

SET START = ptr

SET START->next =NULL

SET START->prev = NULL

SET START->next = START->prev = ptr

Goto Step 10

SET START->prev->next = ptr

SET ptr->prev = START->prev

SET START->prev = ptr

SET ptr->next = START

SET START=ptr

Goto Step 10

Step 3: SET temp = START

Step 4: SET i = 0

Step 5:Repaet step 6&7 while i<position-1 AND temp->next != START

Step 6: SET temp = temp->next

Step 7: SET i++

Step 8: IF temp == NULL

PRINT “Less elements”

Goto Step 10

Step 9: ELSE

[INITIALIZE] ptr ,emp

SET ptr->data= data

SET emp=START

Repeat step while emp->data != val

SET emp = emp->next

SET ptr->next = emp->next

SET ptr->prev = emp

SET emp->next->prev = ptr

SET emp->next = ptr

Step 10:EXIT

b)Before a given value:

Step 1: [INITIALIZE] ptr ,temp

Step 2: SET ptr->data= data

Step 3: SET temp=START

Step 4: IF START == NULL

PRINT “LIST EMPTY”

Goto Step 11

Step 5: Repeat step 6 while temp->data != val

Step 6: SET temp = temp->next

Step 7: SET ptr->next = temp

Step 8: SET ptr->prev = temp->prev

Step 9: SET temp->prev->next = ptr

Step 10: SET temp->prev = ptr

Step 11:EXIT

c)After a given Value:

Step 1: [INITIALIZE] ptr ,temp

Step 2: SET ptr->data= data

Step 3: SET temp=START

Step 4: IF START == NULL

PRINT “LIST EMPTY”

Goto Step 11

Step 5: Repeat step 6 while temp->data != val

Step 6: SET temp = temp->next

Step 7: SET ptr->next = temp->next

Step 8: SET ptr->prev = temp

Step 9: SET temp->next->prev = ptr

Step 10: SET temp->next = ptr

Step 11:EXIT

d)At the beginning

Step 1: [INITIALIZE] ptr

Step 2: SET ptr->data = data

Step 3: IF START == NULL

SET START = ptr

SET START->next =NULL

SET START->prev = NULL

SET START->next = START->prev = ptr

Goto Step 9

Step 4: SET START->prev->next = ptr

Step 5: SET ptr->prev = START->prev

Step 6: SET START->prev = ptr

Step 7: SET ptr->next = START

Step 8: SET START=ptr

Step 9: EXIT

e)At the end

Step 1: [INITIALIZE] ptr,temp

Step 2: SET ptr->data = data

Step 3: IF START == NULL

SET START = ptr

SET START->next = START->prev = ptr

Goto Step 5

Step 4: ELSE

SET temp = START

Repeat step while temp->next != START

temp = temp->next

SET temp->next = ptr

SET ptr ->prev=temp

SET START -> prev = ptr

SET ptr -> next = START

Step 5: EXIT

2.DELETE

a)Value at a particular Position

Step 1: IF START == NULL

PRINT "Linked list is already empty"

Goto Step 12

Step 2: [INITIALIZE] \*temp

Step 3: IF position == 1

[INITIALIZE] ptr

SET ptr=START

SET START = START->next

SET START->prev = NULL

free(ptr)

Step 4: SET temp= START

Step 5: SET i=0

Step 6:Repeat step 7 &8 while i<position-1 && temp!=NULL

Step 7: SET temp = temp->next

Step 8: SET i++

Step 9: IF temp == NULL

PRINT “Less nodes”

Goto step 12

Step 11:ELSE

[INITIALIZE] \*ptr , \*empty

SET ptr= START

Repeat step while ptr != temp->data

SET ptr = ptr->next

SET empty = ptr->next

SET ptr->next = empty->next

SET empty->next->prev = ptr

free(empty)

Step 12:EXIT

b)Before a particular value

Step 1: IF START == NULL

PRINT "Linked list is already empty"

Goto Step 10

Step 2: [INITIALIZE] \*ptr , \*temp

Step 3: SET ptr= START

Step 4:Repeat step 5 while ptr->data != val

Step 5: SET ptr = ptr->next

Step 6: SET temp = ptr->prev

Step 7: SET ptr->prev = temp->prev

Step 8: SET temp->prev->next = ptr

Step 9: free(temp)

Step 10:EXIT

c)After a particular value

Step 1: IF START == NULL

PRINT "Linked list is already empty"

Goto Step 10

Step 2: [INITIALIZE] \*ptr , \*temp

Step 3: SET ptr= START

Step 4:Repeat step 5 while ptr != val

Step 5: SET ptr = ptr->next

Step 6: SET temp = ptr->next

Step 7: SET ptr->next = temp->next

Step 8: SET temp->next->prev = ptr

Step 9: free(temp)

Step 10:EXIT

d)Value at the beginning

Step 1: IF START == NULL

PRINT "List is empty"

Goto Step 8

Step 2: [INITIALIZE] ptr

Step 3: SET ptr=START

Step 4: SET ptr->prev->next = ptr->next

Step 5: SET ptr->next->prev = ptr->prev

Step 6: SET START=START->next

Step 7: free(ptr)

Step 8: EXIT

e)At the end

Step 1: IF START == NULL

PRINT "List is empty"

Goto Step 4

Step 2: ELSE IF START->next == START

SET START = NULL

free(START)

Step 3: ELSE

[INITIALIZE] ptr = START

Repeat while ptr -> next != NULL

SET ptr = ptr->next

SET ptr->prev->next = START

SET START->prev = ptr->prev

free(ptr)

Step 4:EXIT

3.UPDATE

a)Value at a given Position

Step 1: IF START == NULL

PRINT "List is empty"

Goto Step 7

Step 3: IF position == 1

SET START->data = data

Step 4: SET temp= START

Step 5: SET i=0

Step 6:Repeat step 7 &8 while i<position-1 && temp!=NULL

Step 7: SET temp = temp->next

Step 8: SET i++

Step 9: IF temp == NULL

PRINT “Less nodes”

Goto step 12

Step 11:ELSE

[INITIALIZE] \*ptr=START

Repeat step while ptr->data != temp->data

SET ptr = ptr->next

SET ptr->next->data = data

Step 12:EXIT

b)Before a particular value

Step 1: IF START == NULL

PRINT "List is empty"

Goto Step 6

Step 2: [INITIALIZE] \*ptr=START

Step 3:Repeat step 4 while ptr->data != val

Step 4: ptr = ptr->next

Step 5: SET ptr->prev->data = data

Step 6:EXIT

c)After a particular value

Step 1: IF START == NULL

PRINT "Linked list is already empty"

Goto Step 6

Step 2: [INITIALIZE] \*ptr=START

Step 3:Repeat step 4 while ptr->data != val

Step 4: SET ptr = ptr->next

Step 5:SET ptr->next->data = data

Step 6: EXIT

d)Value at the beginning

Step 1: IF START == NULL

PRINT "List is empty"

Goto Step 3

Step 2: SET START->data = data

Step 3:EXIT

e)At the end

Step 1: IF START == NULL

PRINT "Linked list is already empty"

Goto Step 4

Step 2: [INITIALIZE] \*ptr=START->prev

Step 3:SET ptr->data = data

Step 4:EXIT

4.SEARCH

Step 1: IF START == NULL

PRINT "List is empty"

Goto Step 9

Step 2: [INITIALIZE] \*ptr = START

Step 3: SET count =1

Step 4: Repeat Step 5&6 while ptr->data != data && count<=countNodes()+1

Step 5: SET ptr = ptr->next

Step 6: SET count=1

Step 7: IF count>countNodes()

PRINT ‘ELEMENT NOT FOUND’

Step 8: ELSE

PRINT “Element found at the position”

Step 9: EXIT

5.COUNT NODES

Step 1:[ INITIALIZE] \*ptr = START

Step 2: SET count = 1

Step 3:Repeat step 4&5 while ptr->next !=NULL

Step 4: SET count++

Step 5: SET ptr=ptr->next

Step 6 :RETURN count

Step 7:EXIT

7.DISPLAY

Step 1:[INITIALIZE] \*ptr =START

Step 2:Repeat step 3&4 while ptr != NULL

Step 3: PRINT (ptr->data)

Step 4: SET ptr = ptr->next

Step 5: PRINT (ptr->data)

Step 6: EXIT