Doubt Clearing Session

Course on C-Programming & Data Structures: GATE - 2024 & 2025

Vishvadeep Gothi • Lesson 31 • Jan 6, 2023

Data Structure:
Doubts & Queue

By: Vishvadeep Gothi

Hello!

I am Vishvadeep Gothi

I am here because I love to teach

DPP

Question 1

Write an algorithm to convert the list into a circular list?

Question 2

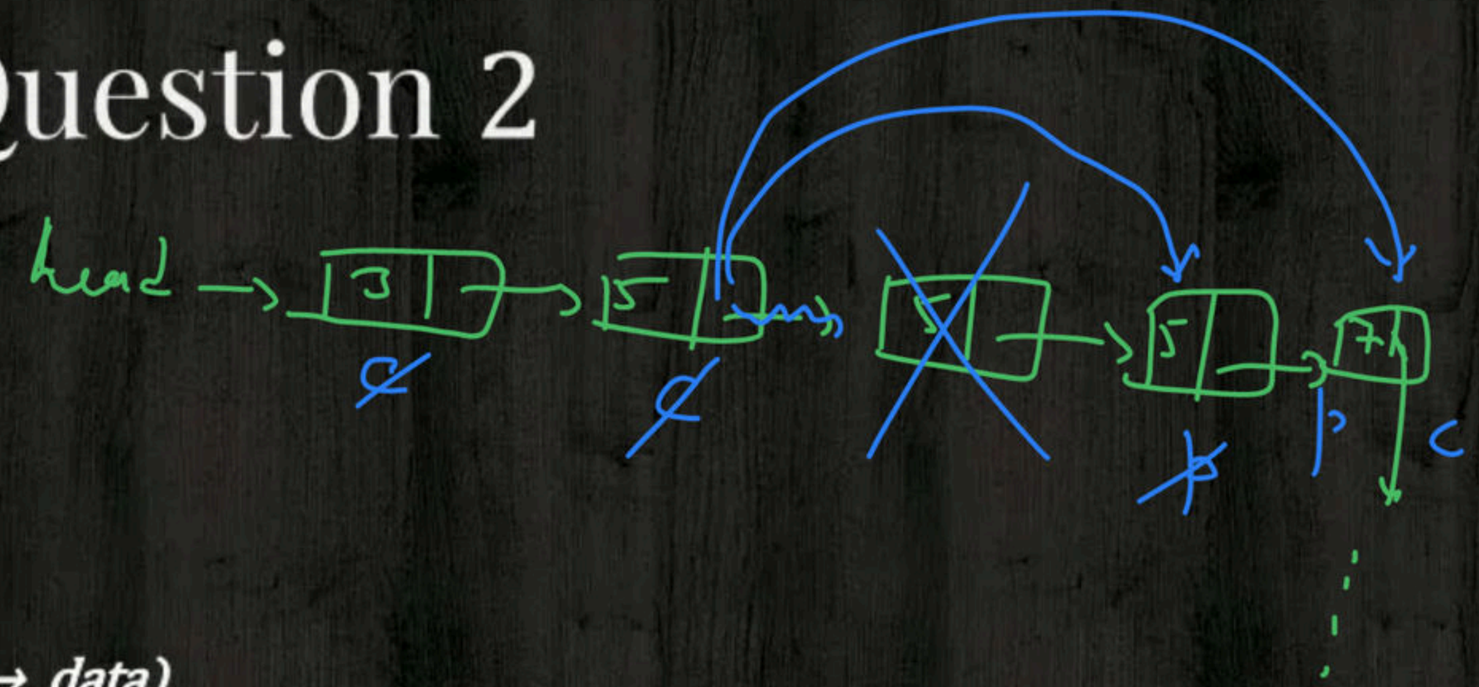
Following C- like function takes a singly-linked list of integers as a parameters and rearranges the elements of list. The function is called with the list containing the integers 3, 5, 5, 5, 7, 8, 9, 9, 9, 9, 12, 15, 18, 19, 23 in the given order. What will be the total no. of contents of the list after the function completes execution (no. of nodes in the list)

Ans = 16

```
struct node
{
 int data;
struct node * next;
};
```

Question 2

```
void rearrange (struct node * head)
{
struct node * current = head;
if( current = = NULL) return;
while (current →next ! = NULL)
{
   if (current →data = = current → next→ data)
      {
      struct node *P = current → next→next;
      free (current → next );
      current → next = p;
      }
   else
   {
      current = current → next;
   }}}
```



Question 3

Consider the following function on a valid NULL terminated list:

```
int func(node *start)
struct node *p;
p = start ;
while(p → link )
{
p  = p→ link;
}
return 1;
```

The above function returns:

(A) 1 always

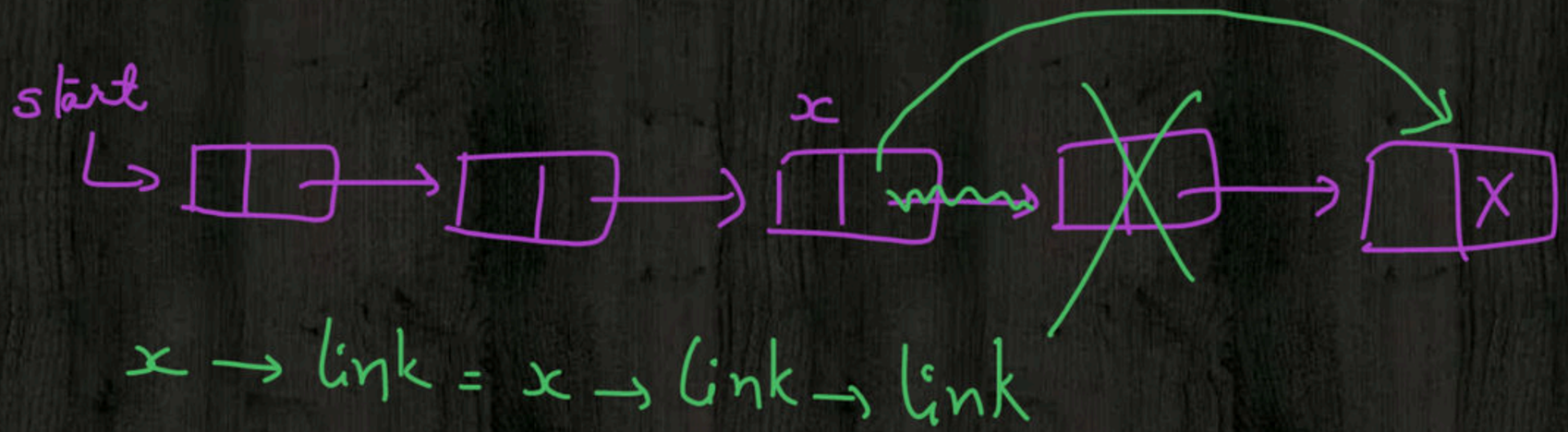
(B) None always

(C) Will cause error always

(D) Error or returns 1

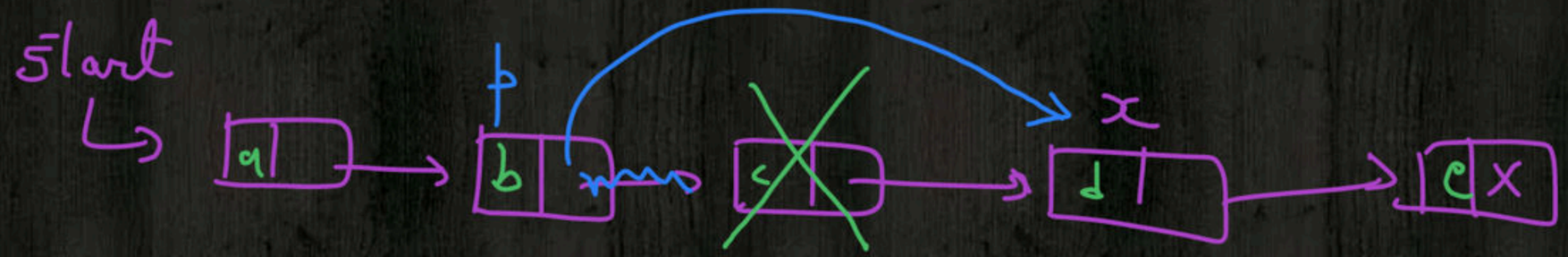
Question 4

There is node pointer x, which points to a node of a singly linked list. Delete the node after the node pointed by pointer x.



Question 5

There is node pointer x, which points to a node of a singly linked list. Delete the node before the node pointed by pointer x.

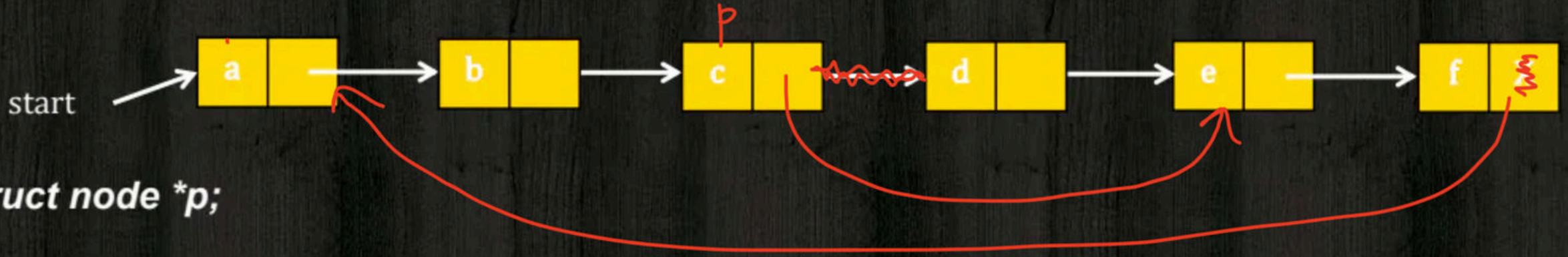



```
p = start
while( p -> link -> link != x)
{ p = p -> link
} p -> link = x
```

O(n)

Question 6

What would be the output after the sequence of steps:



struct node *p;

p =start→ link → link;

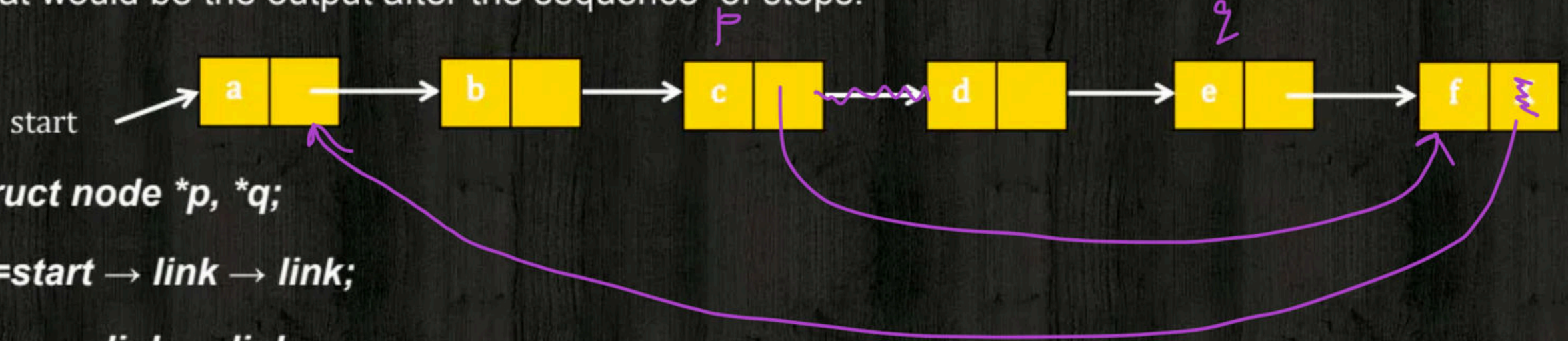
start → link → link → link = p → link → link;

p → link → link → link = start;

printf("%c", start → link → link → link → link → data); f

Question 7

What would be the output after the sequence of steps:



struct node *p, *q;

p =start → link → link;

q = p → link → link;

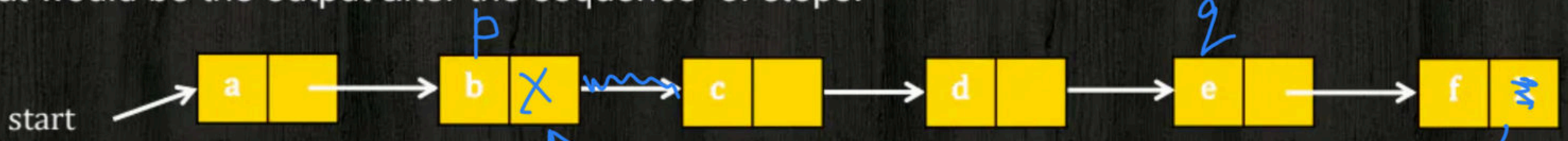
p → link = q → link;

q → link → link = start;

printf("%c", start → link → link → link → link → data);

Question 8

What would be the output after the sequence of steps:



struct node *p, *q;

p =start → link;

q = p → link → link → link;

p → link = q → link → link;

q → link → link = p;

printf("%c", start → link → link → link → link → data);

NULL

NPJ

Question 1

Given a singly linked list, check whether there is any loop in the list or not?

Question 2

Find the intersection point of a Y-shaped List?

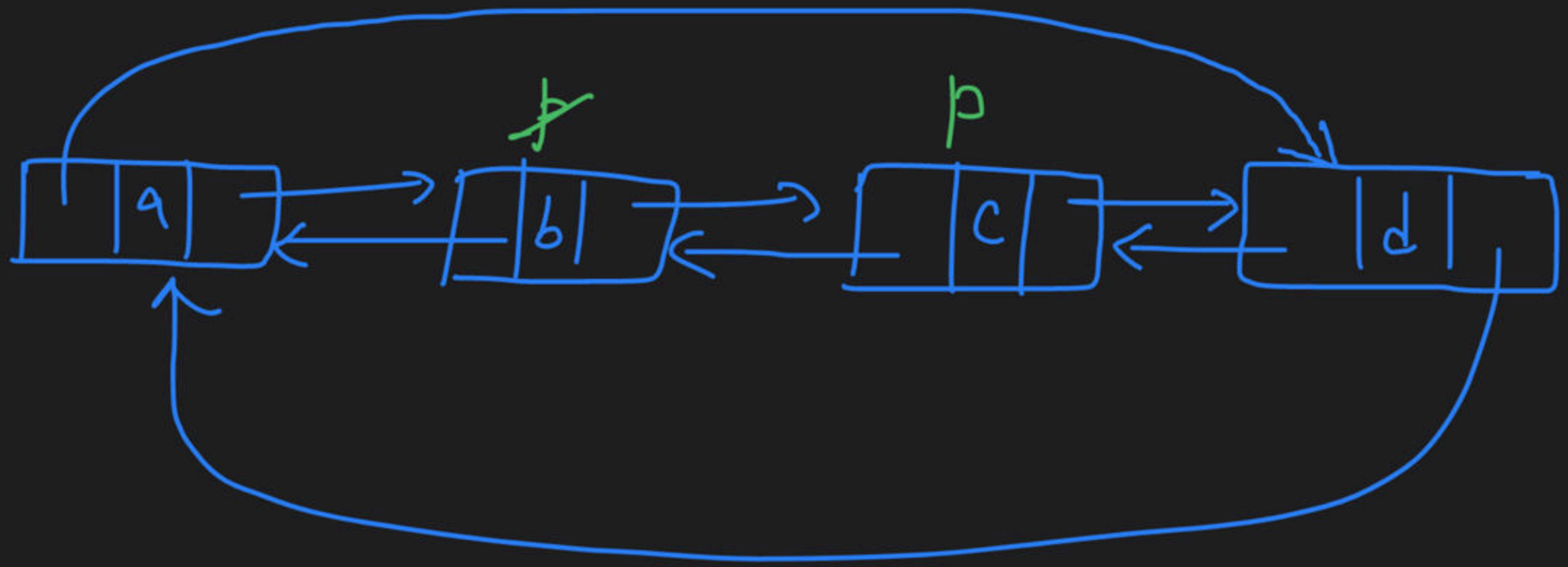
Question 3

Write an algorithm to reverse a singly linked list?

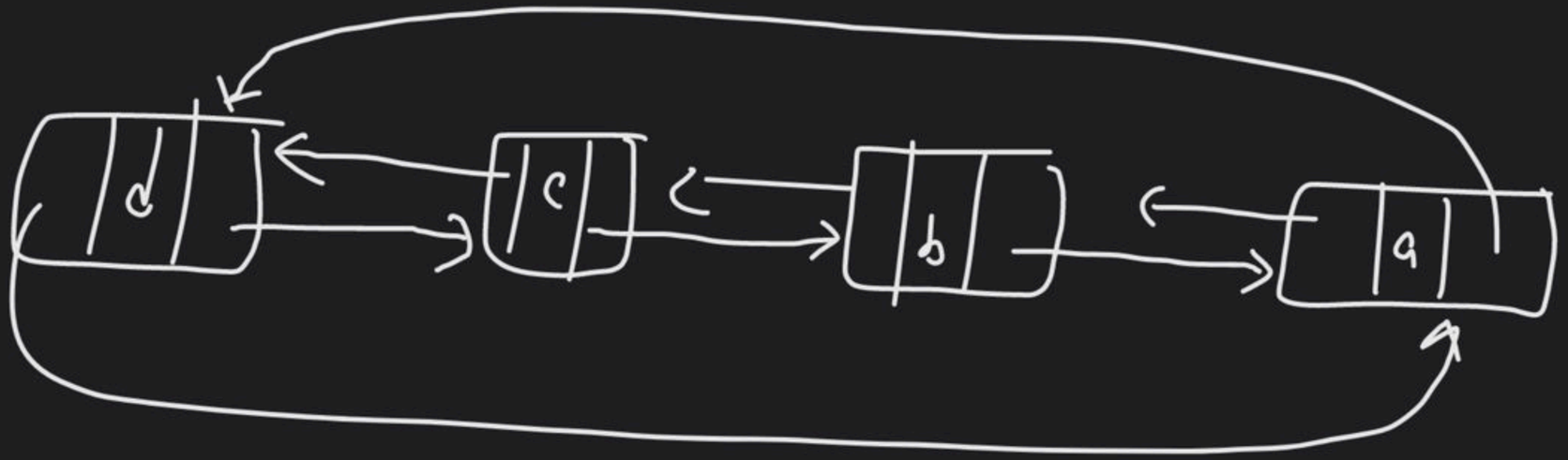
Question 1

- How many links updated when insertion and deletion done in doubly linked list?

Doubly circular linked list :-



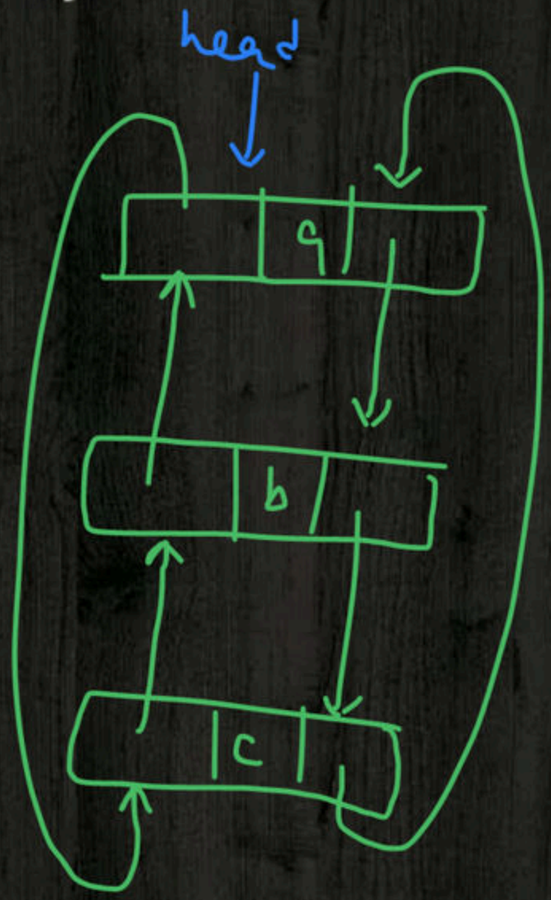
Reverse

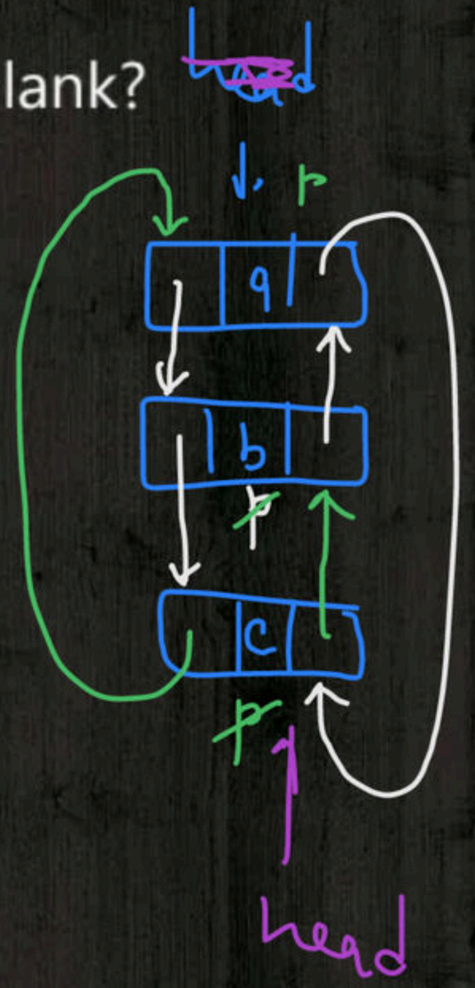


Question 2

The following code is given to reverse the doubly linked list. Fill the blank?

```
void reverse(Dnode *head)
{
struct Dnode *p = head;
while(p != head)
{
swap(p → forw, p → back);
p = p → back;
}
swap( head → forw, head → back)
head = head → forw;
}
```





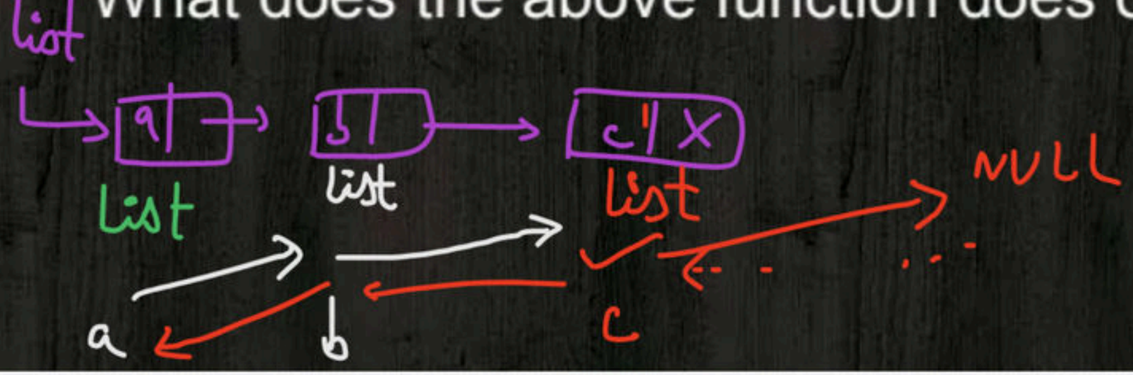
Question 3

```
void fun(struct node * list)
{
 if(! list)
        return;

 fun(list → link);
 printf("%c", list → data);
}
```

c b a

prints elements of LL in reverse order

What does the above function does on a singly linked list?

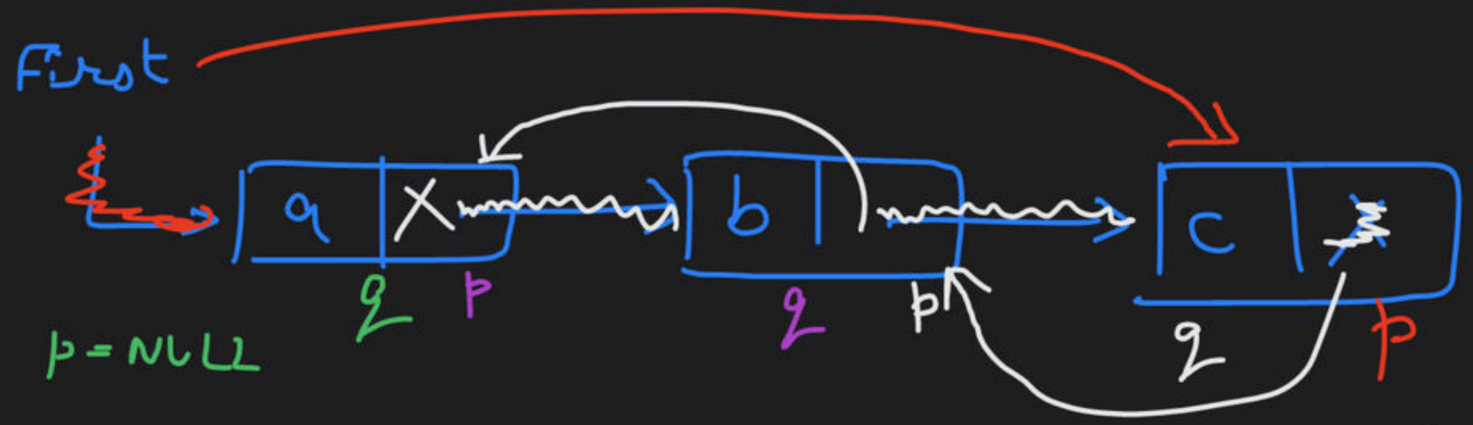


Question 4

What does fun() function do, if called as fun(NULL, First), where First is a list pointer, pointing to the first node of a singly list?

```
void fun(struct node * p, struct node *q)
{
  if(q)
    {
       fun(q, q → link);
       q → link =p;
    }
    else
        First = p;
}
```

(A) No changes in the list
(B) Reverses the list
(C) Swaps every consecutive pair of nodes
(D) Deletes last node and adds it to front



fun(NULL, First)
{
fun(2, 2→link)
2→link = p
}

fun( )
{
fun(2, 2→link)
2→link = p
}

fun( )
{
fun(2, 2→link)
2→link = p
}

First=p

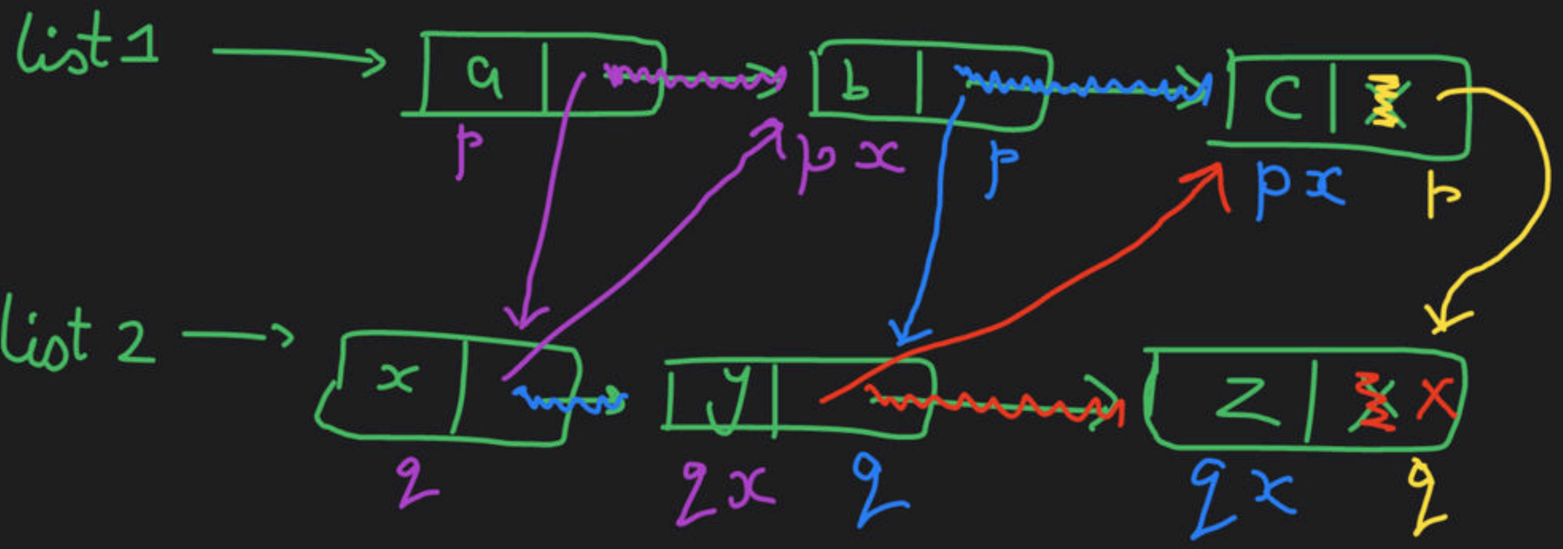
First


Question 5

What does fun() function do, if called with list pointers of 2 singly linked lists?

```
void fun(struct node * p, struct node *q)
{
struct Node *px, *qx;
if(!p)
   return q;
else if(!q)
   return p;
else
{
  px = p → link;
 qx = q → link;
 p → link = q;
 q → link = fun(px, qx);
 return p;
}
```

(A) Concatenation of 2 lists by selecting alternate nodes
(B) Append list p at the end of list q for all inputs
(C) Append list q at the end of list p for all inputs
(D) Append reverse of list p at the end of list q for all inputs




```
fun(p, q)
{
q->link = fun(px, qx)
return p
}
```

q->link = fun(px, qx)

return p;

q->link = fun(px, qx)

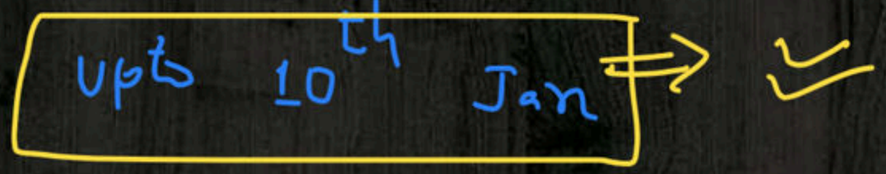
return p;

final

Question 6

```
void fun(struct node * list)
{
  if(! list → link)
                return list;

  return fun(list → link);
}
```

What does the above function does on a singly linked list?



Data Structure:
PYQ: Linked List

By: Vishvadeep Gothi

Question GATE-1987

In a circular linked list oraganisation, insertion of a record involves modification of

A. One pointer.
B. Two pointers.
C. Multiple pointers.
D. No pointer.

Question GATE-1993

Consider a singly linked list having n nodes. The data items $d_1, d_2, \ldots d_n$ are stored in these n nodes. Let X be a pointer to the j^{th} node $(1 \leq j \leq n)$ in which d_j is stored. A new data item d stored in node with address Y is to be inserted. Give an algorithm to insert d into the list to obtain a list having items $d_1, d_2, \ldots, d_j, d, \ldots, d_n$ in order without using the header.

Question GATE-1994

Linked lists are not suitable data structures for which one of the following problems?

A. Insertion sort

B. Binary search

C. Radix sort

D. Polynomial manipulation

Question GATE-1995

Which of the following statements is true?

I. As the number of entries in a hash table increases, the number of collisions increases.

II. Recursive programs are efficient

III. The worst case complexity for Quicksort is $O(n^2)$

IV. Binary search using a linear linked list is efficient

A. I and II B. II and III C. I and IV D. I and III

Question GATE-1997

The concatenation of two lists is to be performed on $O(1)$ time. Which of the following implementations of a list should be used?

A. Singly linked list

B. Doubly linked list

C. Circular doubly linked list

D. Array implementation of list

Question GATE-1997

Consider the following piece of 'C' code fragment that removes duplicates from an ordered list of integers.

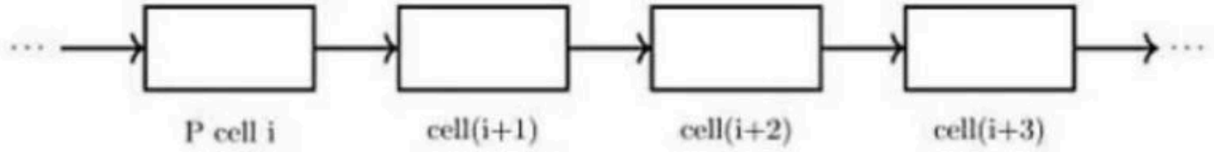
```
Node *remove-duplicates (Node* head, int *j)
{
    Node *t1, *t2; *j=0;
    t1 = head;
    if (t1! = NULL)
        t2 = t1 ->next;
    else return head;
    *j = 1;
    if(t2 == NULL) return head;
    while (t2 != NULL)
    {
        if (t1.val != t2.val) ----------------> (S1)
        {
            (*j)++;
            t1 -> next = t2;
            t1 = t2; -----> (S2)
        }
            t2 = t2 ->next;
    }
    t1 -> next = NULL;
    return head;
}
```

Assume the list contains n elements ($n \geq 2$) in the following questions.

a. How many times is the comparison in statement $S1$ made?
b. What is the minimum and the maximum number of times statements marked $S2$ get executed?
c. What is the significance of the value in the integer pointed to by j when the function completes?

Question GATE-1998

a. Let p be a pointer as shown in the figure in a single linked list.



What do the following assignment statements achieve?

```
q: = p -> next
p -> next:= q -> next
q -> next:=(q -> next) -> next
(p -> next) -> next:= q
```

Question GATE-1999

Write a constant time algorithm to insert a node with data D just before the node with address p of a singly linked list.

Question GATE-2002

In the worst case, the number of comparisons needed to search a single linked list of length n for a given element is

A. $\log n$ B. $\frac{n}{2}$ C. $\log_2 n - 1$ D. n

Question GATE-2003

Consider the function f defined below.

```
struct item {
        int data;
        struct item * next;
};
int f(struct item *p) {
    return ((p == NULL) || (p->next == NULL)||
        ((p->data <= p ->next -> data) &&
        f(p->next)));
}
```

For a given linked list p, the function f returns 1 if and only if

A. the list is empty or has exactly one element

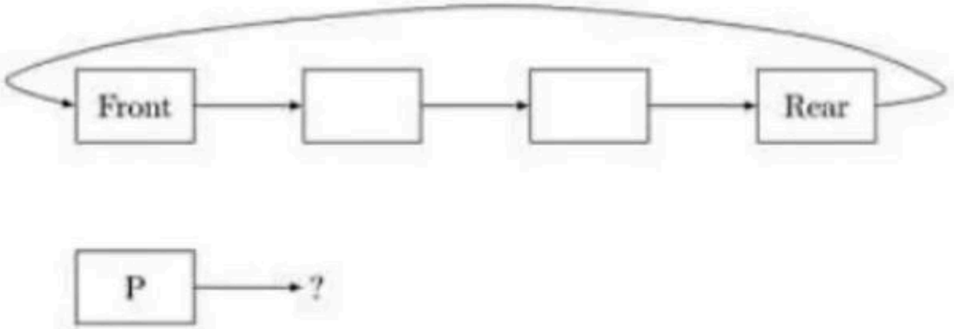
B. the elements in the list are sorted in non-decreasing order of data value

C. the elements in the list are sorted in non-increasing order of data value

D. not all elements in the list have the same data value

Question GATE-2004

A circularly linked list is used to represent a Queue. A single variable p is used to access the Queue. To which node should p point such that both the operations enQueue and deQueue can be performed in constant time?



A. rear node
B. front node
C. not possible with a single pointer
D. node next to front

Question GATE-2004

Suppose each set is represented as a linked list with elements in arbitrary order. Which of the operations among union, intersection, membership, cardinality will be the slowest?

A. union only

B. intersection, membership

C. membership, cardinality

D. union, intersection

Question GATE-2005

The following C function takes a singly-linked list of integers as a parameter and rearranges the elements of the list. The list is represented as pointer to a structure. The function is called with the list containing the integers $1,2,3,4,5,6,7$ in the given order. What will be the contents of the list after the function completes execution?

```
struct node {int value; struct node *next;};
void rearrange (struct node *list) {
    struct node *p, *q;
    int temp;
    if (!list || !list -> next) return;
    p = list; q = list -> next;
    while (q) {
        temp = p -> value;
        p -> value = q -> value;
        q -> value = temp;
        p = q -> next;
        q = p ? p -> next : 0;
    }
}
```

A. $1,2,3,4,5,6,7$
B. $2,1,4,3,6,5,7$
C. $1,3,2,5,4,7,6$
D. $2,3,4,5,6,7,1$

Question GATE-2010

The following C function takes a singly-linked list as input argument. It modifies the list by moving the last element to the front of the list and returns the modified list. Some part of the code is left blank.

```
typedef struct node
{
    int value;
    struct node *next;
} node;
Node *move_to-front(Node *head)
{
    Node *p, *q;
    if ((head == NULL) || (head -> next == NULL))
        return head;
    q = NULL;
    p = head;
    while (p->next != NULL)
    {
        q=p;
        p=p->next;
    }
    ____________________

    return head;

}
```

Choose the correct alternative to replace the blank line.

A. $q = NULL; p \rightarrow next = head; head = p$;
B. $q \rightarrow next = NULL; head = p; p \rightarrow next = head$;
C. $head = p; p \rightarrow next = q; q \rightarrow next = NULL$;
D. $q \rightarrow next = NULL; p \rightarrow next = head; head = p$;

Question GATE-2016

N items are stored in a sorted doubly linked list. For a *delete* operation, a pointer is provided to the record to be deleted. For a *decrease-key* operation, a pointer is provided to the record on which the operation is to be performed.

An algorithm performs the following operations on the list in this order: $\Theta(N)$ *delete*, $O(\log N)$ *insert*, $O(\log N)$ *find*, and $\Theta(N)$ *decrease-key*. What is the time complexity of all these operations put together?

A. $O(\log^2 N)$ B. $O(N)$ C. $O(N^2)$ D. $\Theta(N^2 \log N)$

Question GATE-2017

Consider the C code fragment given below.

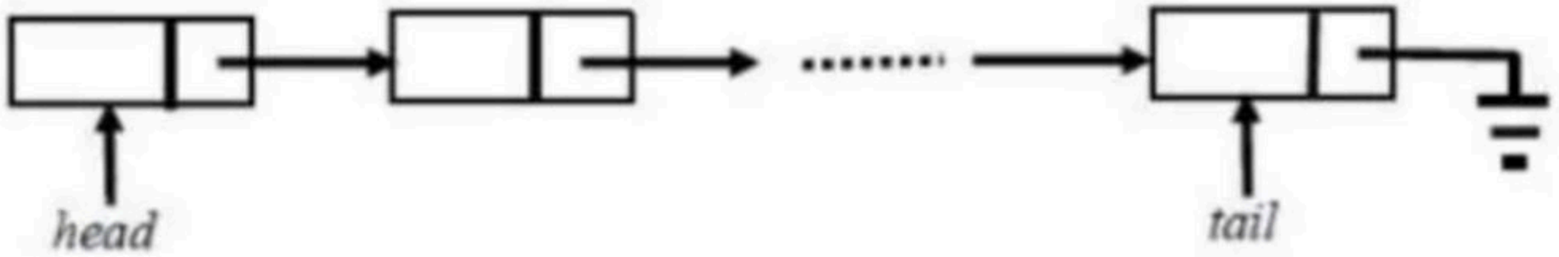
```
typedef struct node {
    int data;
    node* next;
} node;

void join(node* m, node* n) {
    node* p = n;
    while(p->next != NULL) {
        p = p->next;
    }
    p->next = m;
}
```

Assuming that m and n point to valid NULL-terminated linked lists, invocation of join will

A. append list m to the end of list n for all inputs.
B. either cause a null pointer dereference or append list m to the end of list n.
C. cause a null pointer dereference for all inputs.
D. append list n to the end of list m for all inputs.

Question GATE-2018

A queue is implemented using a non-circular singly linked list. The queue has a head pointer and a tail pointer, as shown in the figure. Let n denote the number of nodes in the queue. Let 'enqueue' be implemented by inserting a new node at the head, and 'dequeue' be implemented by deletion of a node from the tail.



Which one of the following is the time complexity of the most time-efficient implementation of 'enqueue' and 'dequeue, respectively, for this data structure?

(A) Θ(1), Θ(1)

(B) Θ(1), Θ(n)

(C) Θ(n), Θ(1)

(D) Θ(n), Θ(n)

Question GATE-2020

What is the worst case time complexity of inserting n elements into an empty linked list, if the linked list needs to be maintained in sorted order ?

(A) $\Theta(n)$

(B) $\Theta(n \log n)$

(C) $\Theta(n^2)$

(D) $\Theta(1)$

Question GATE-2021

Consider the following ANSI C program:

```
#include <stdio.h>
#include <stdlib.h>
struct  Node{
        int value;
        struct Node *next;};
int main(){
   struct Node  *boxE, *head, *boxN; int index = 0;
   boxE = head = (struct Node *) malloc(sizeof(struct Node));
   head->value = index;
   for (index = 1; index <= 3; index++){
        boxN = (struct Node *) malloc(sizeof(struct Node));
        boxE->next =  boxN;
        boxN->value = index;
        boxE = boxN; }
   for (index = 0; index <= 3;  index++) {
        printf("Value at index  %d is %d\n", index, head->value);
        head = head->next;
        printf("Value at index %d is %d\n", index+1, head->value); } }
```

Which one of the following statements below is correct about the program?

A. Upon execution, the program creates a linked-list of five nodes
B. Upon execution, the program goes into an infinite loop
C. It has a missing **return** which will be reported as an error by the compiler
D. It dereferences an uninitialized pointer that may result in a run-time error

Happy Learning