Linked List: Basics and Oprations

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

Vishvadeep Gothi • Lesson 25 • Dec 24, 2022

Data Structure:
Linked List 1

By: Vishvadeep Gothi

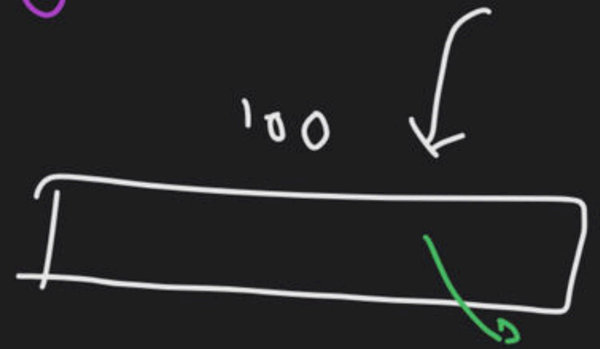
Hello!

I am Vishvadeep Gothi

I am here because I love to teach

Array:- advantages

→ ① Random or direct access, using index



Disadv. :-

→ ① fixed size (predefined size)

↳ not very flexible with no. of elements increases

→ ② stored on consecutive locat^n

sol^n

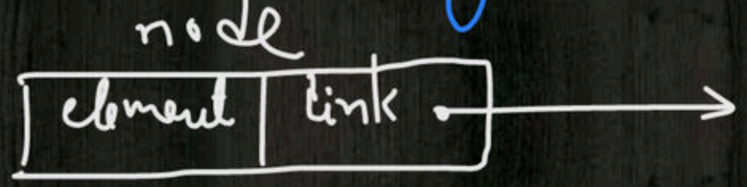
↳ linked list

Linked List

→ Linear D.S.

→ Linear order is maintained using pointers or links.

→ LL. contains nodes



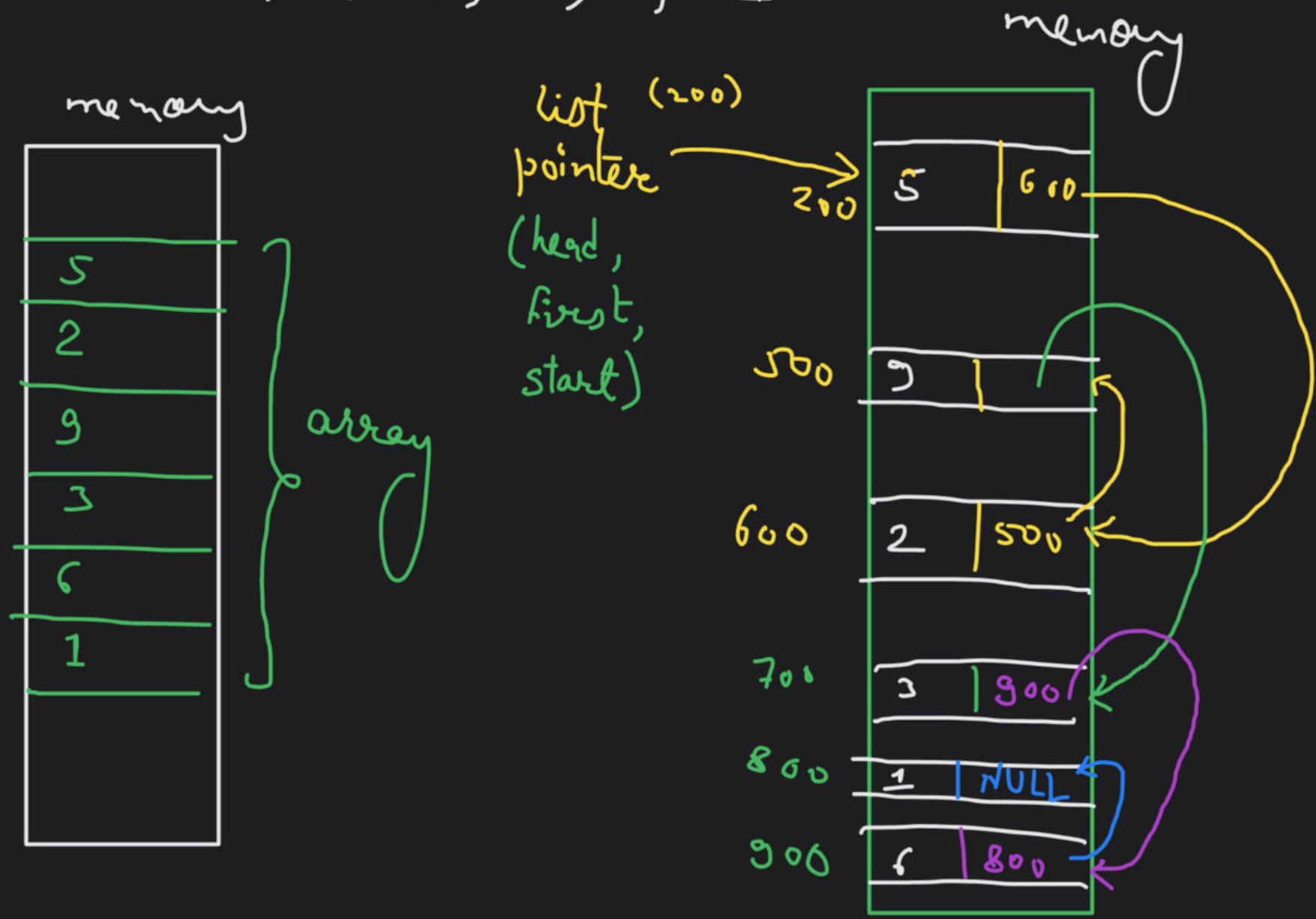
→ element

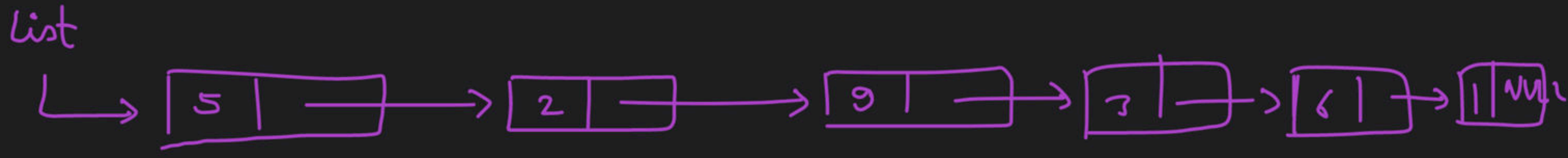
→ link or address of next node

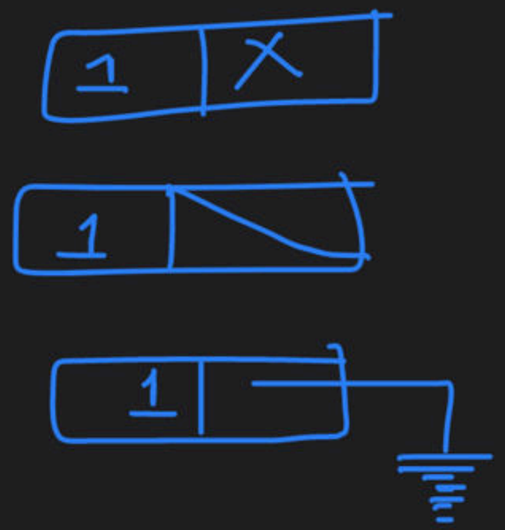
→ Last node contains NULL in link part; to represent end of the list

→ Add. of first node is stored in a list pointer.

elements :- 5, 2, 9, 3, 6, 1




list
5
2
9
7
6
1

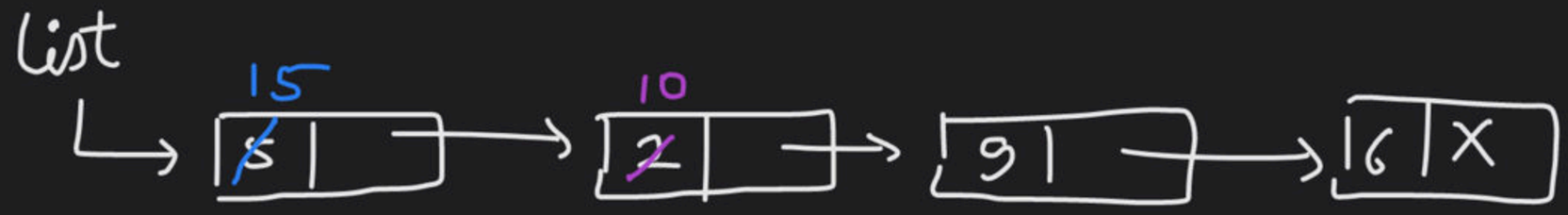

1
X
1
1

Node Structure

```
struct node
{
  int data;          (element, item, value, key)
  struct node *link;  (next, forward, rightpointer)
};
```

self referential structure

Accessing members of node:-




```
printf("%d", list -> data);   5

list -> data = 15;

printf("%d", list -> link -> data);   2

list -> link -> data = list -> data - 5;

list -> link = NULL;
```

Empty list:-

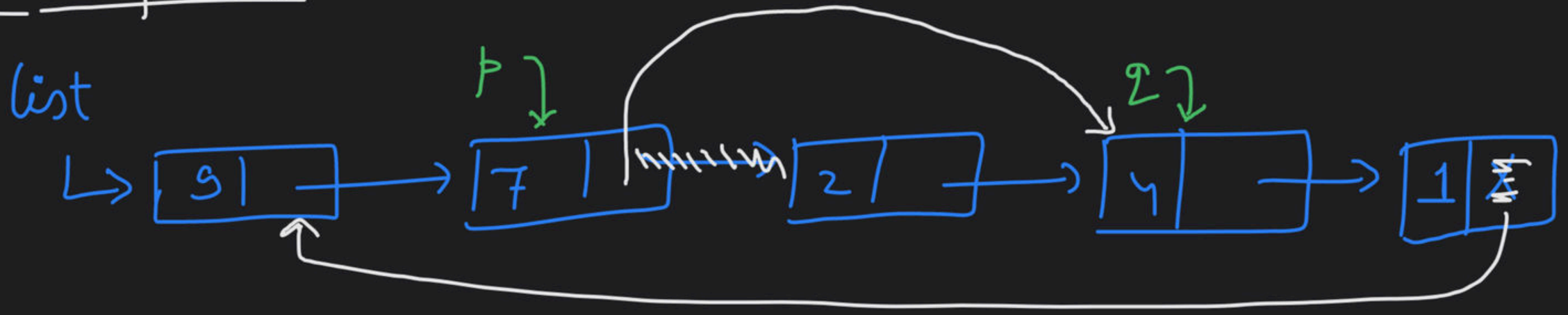
if (list == NULL) or if (!list)

Single node:-

list

if (list → link == NULL) or if (!list → link)

Link updation :-




```
p = list -> link;
q = p -> link -> link;
p -> link = q;
q -> link -> link = list;
printf("%d", q -> link -> link -> link -> data); 7
```

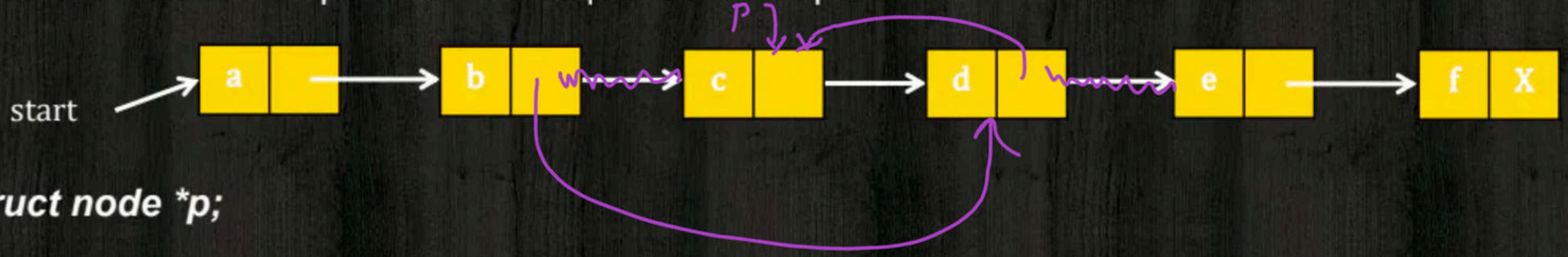
assignment rule :-

(a) = (b) value.

location

Practice Question 1

What would be the output after the sequence of steps:



struct node *p;

p =start→ link → link

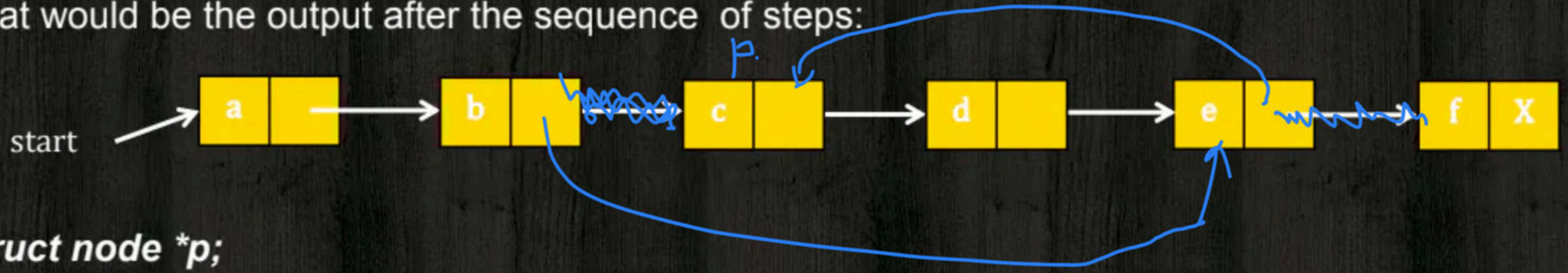
start → link → link = p → link

p → link → link = p

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

Practice Question 2

What would be the output after the sequence of steps:



struct node *p;

p = start → link → link

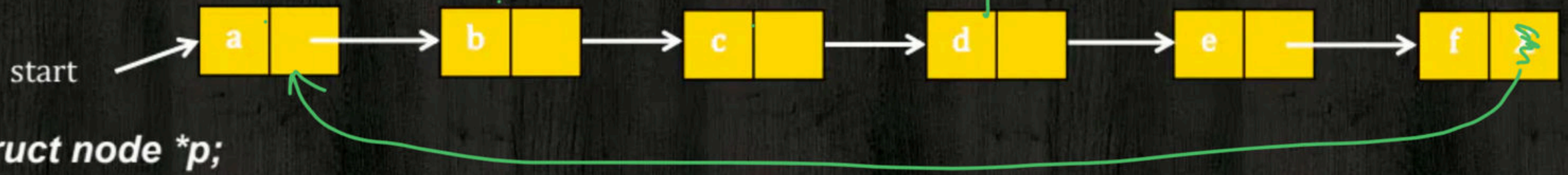
start → link → link = p → link → link

p → link → link → link = p

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

Practice Question 3

What would be the output after the sequence of steps:



struct node *p;

p = start → link → link → link ;

(p→ link → link)→ link = start;

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

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

Traversing in Linked List

Number of Elements in Linked List

Address of Last Node in Linked List

Question

What would be the output after the following code segment is executed on a valid NULL terminated singly linked list.

```
struct node *p;
p = start ;
while(p → link → link)
{
p  = p→ link;
}
printf("%c", p → data );
```

Insertion in Linked List

1. Insertion at beginning
2. Insertion after given node
3. Insertion at the end

Insertion at beginning

Insertion After Given Node

Insertion At the End

Deletion in Linked List

1. Deletion at beginning
2. Deletion of given node
3. Deletion at the end

Deletion At Beginning

Deletion Of a Given Node

Deletion At the End

Question 1 GATE-2004

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

(A) Union only

(B) Intersection, membership

(C) Membership, cardinality

(D) Union, intersection

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)

```
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.

Happy Learning