unacademy

Linked List - Part V

Course on Data Structure

Pankaj Sharma • Lesson 11 • July 19, 2024

unacademy

CS & IT Engineering

Data Structure

Linked List- IV

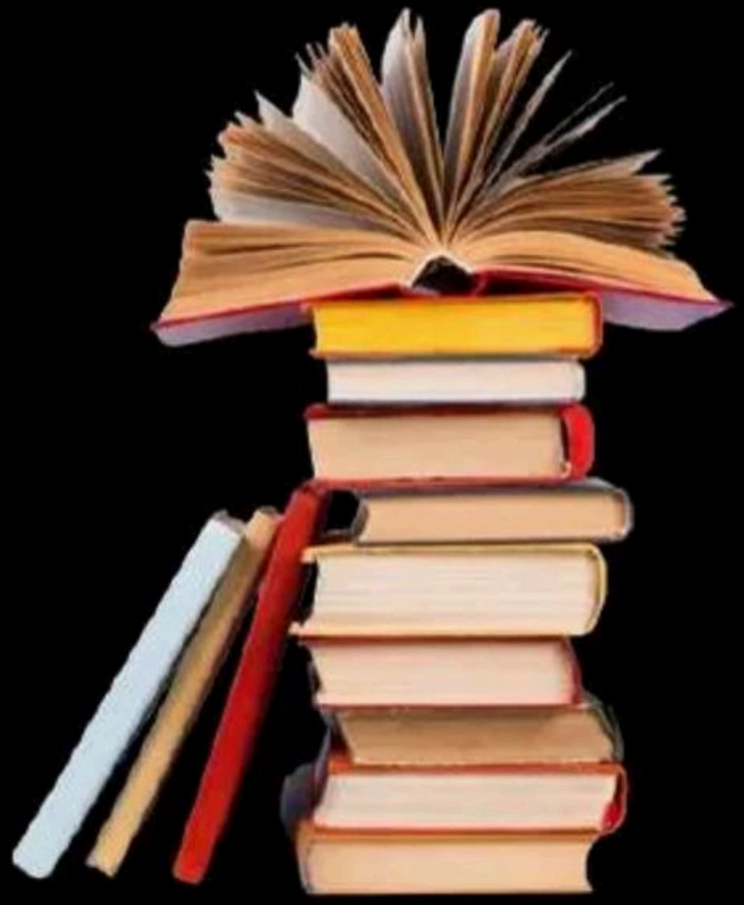
Lecture Number- 03

By- Pankaj Sir

unacademy

Topics

to be covered



1. Linked List

4. Given a linked list print 2nd last node data.

(2-3 min)

wrong →

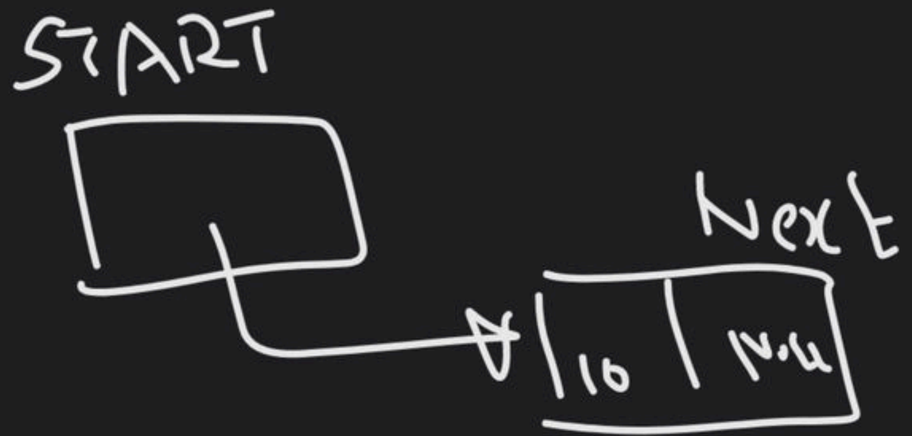
```
Ptr = START;
[
  while( Ptr → Next → Next != NULL)
  [
    Pt = Ptr → Next;
```

Ensured that atleast 2 nodes ⇒

unacademy

L.L. Empty $\Rightarrow$ 0 Node $\Rightarrow$ START == NULL

LL. only 1 node

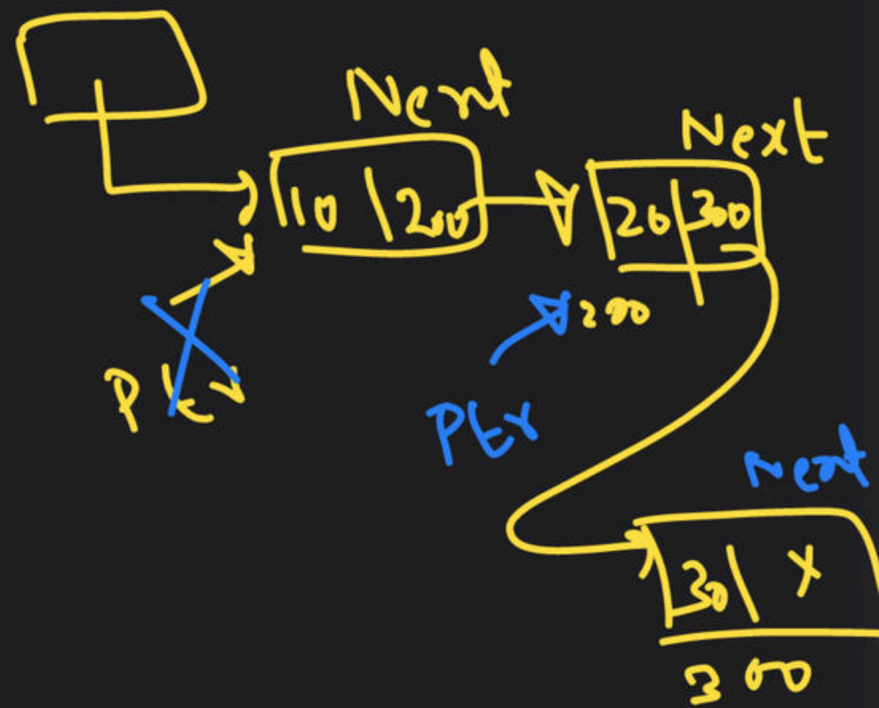


{START → Next == NULL}

```
struct Node *ptr;
if(START==NULL || START->Next==NULL)
        return;
```

→ `Ptr = START;`

```
while(Ptr->Nxt->Next!=NULL)
        Ptr = Ptr->Next;
printf("%d",Ptr->data);
```



```
void Traversal (struct Node * ptr)
{
    ...
    ...
    ...
    ...
}
```

```
void main(){
struct Node *START;
    ...
    ...
Traversal(START);
    ...
    ...
}
```

unacademy

▲ 1 • Asked by Jaiprakash

Sir maine kal ke hw ka code try kiya

```
void search(struct node* start,int target){
    // If LL empty
    if(start==NULL){
        printf("LL empty");
        return;
    }

    // If not empty
    struct node* ptr=start;
    int pos=1;
    while(ptr!=NULL){
        if(ptr->data==target){
            printf("Target found at position %d",pos);
            return;
        }
        pos++;
        ptr=ptr->next;
    }

    // means element not found in LL
    printf("Target not found");
}
```

Given a LL & a key. find whether the key is present in LL or not.

key: 50

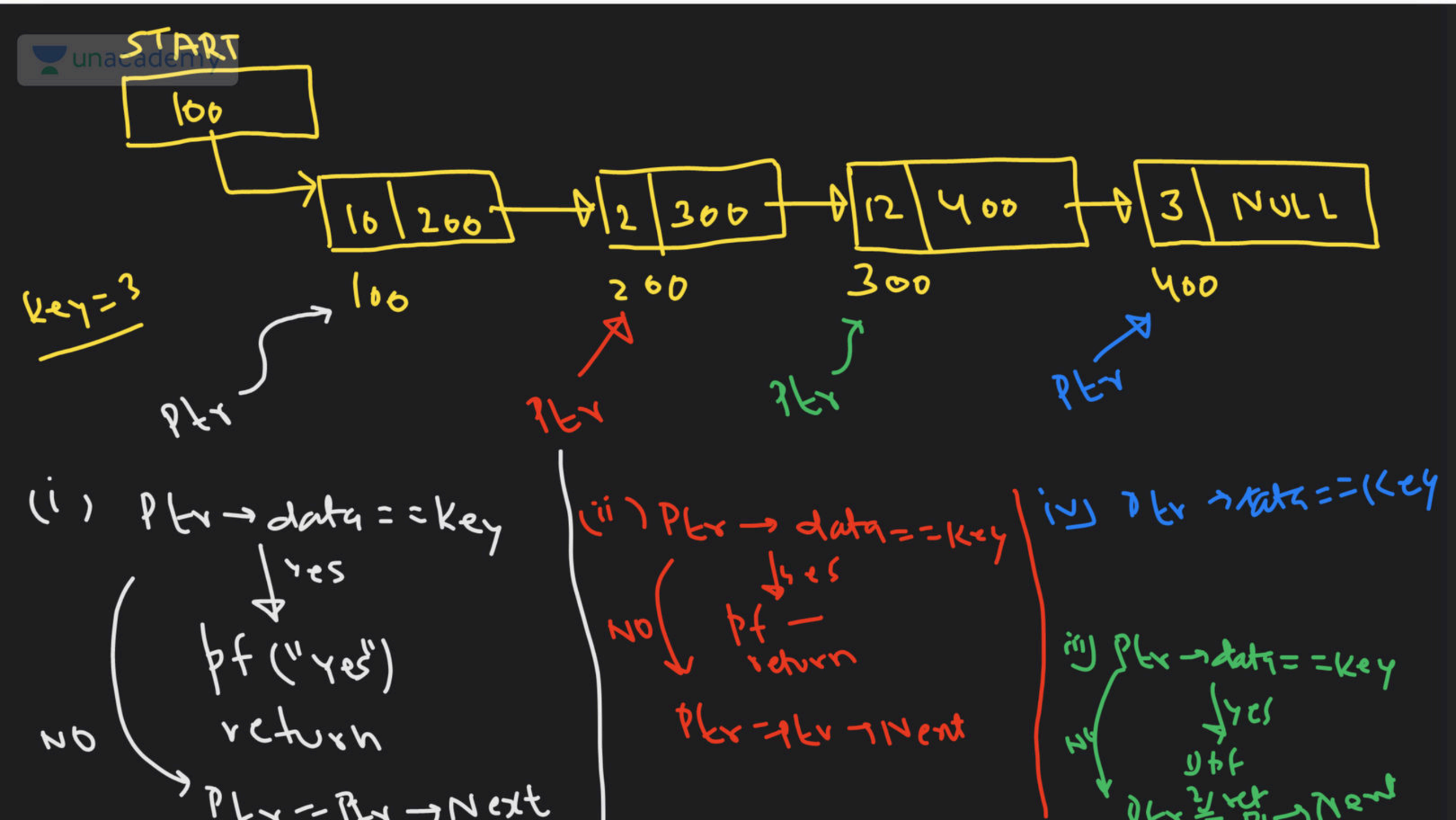
o/p: No

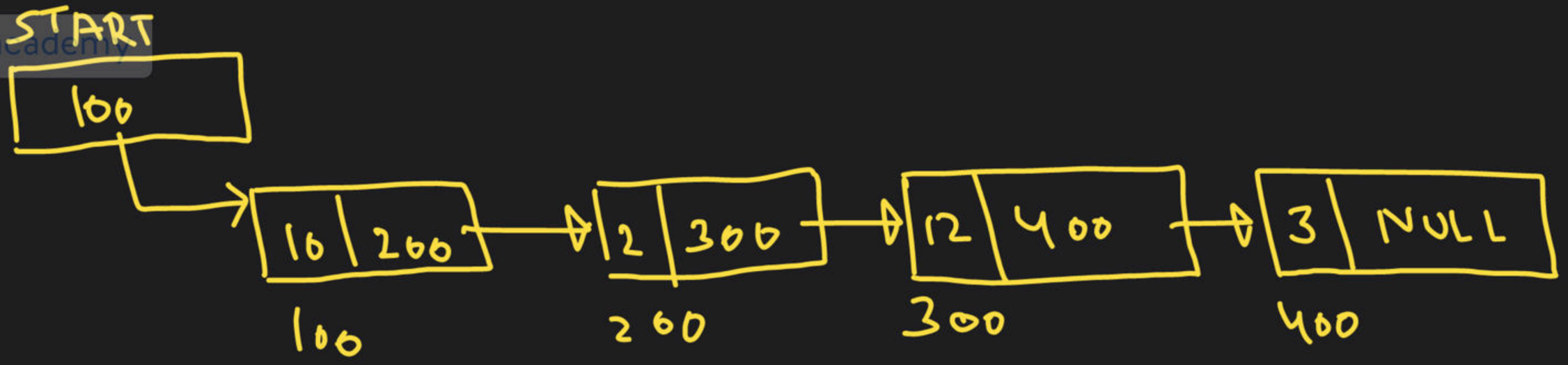
Key: 10

o/p: Yes

```
void Search(int key)
{

}
```

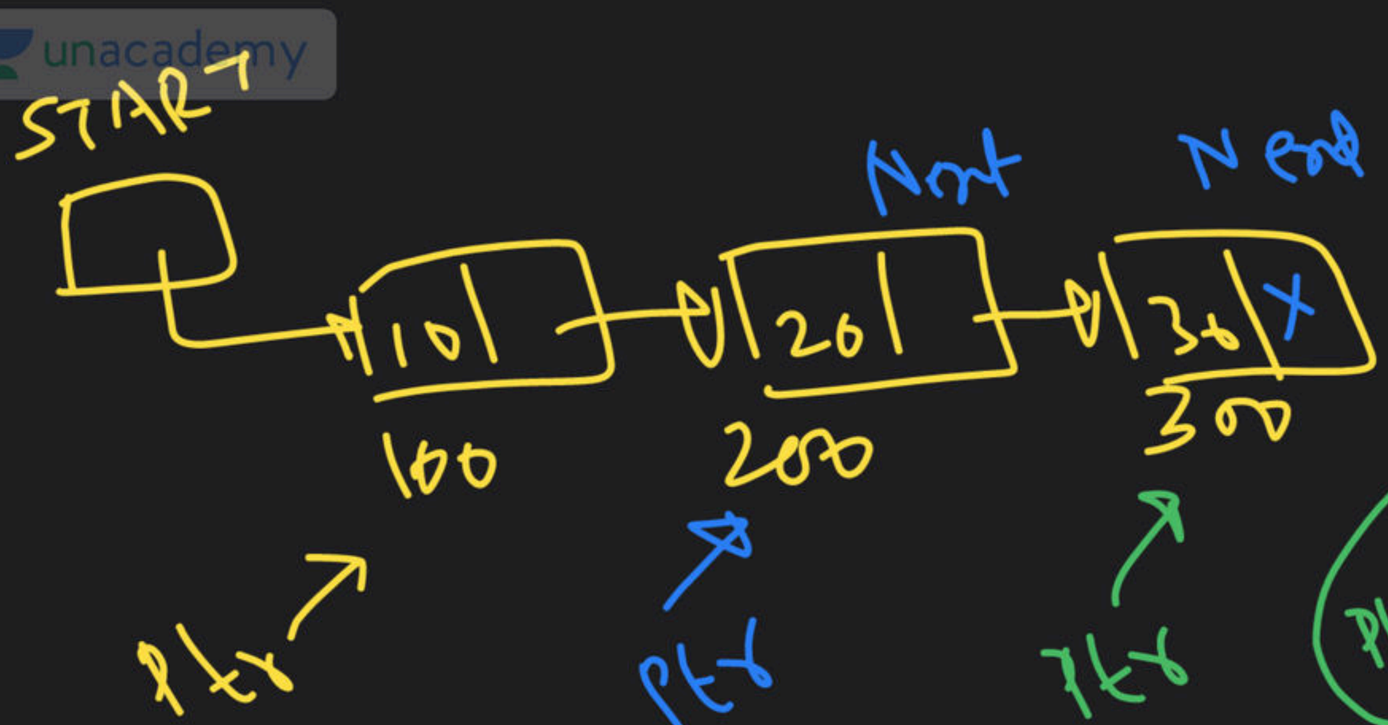
```
void search(struct Node *Ptr,
            int key){

}

void main(){
search(START,100);
}
```


START
100
10 | 200
2 | 300
12 | 400
3 | NULL
100
200
300
400
Key=3
Ptr
Ptr
Ptr
Ptr
(i) Ptr → data == key
yes
pf ("Yes")
return
NO
Ptr = Ptr → Next
(ii) Ptr → data == key
yes
pf —
return
NO
Ptr = Ptr → Next
(iii) Ptr → data == key
yes
pf
NO
(iv) Ptr → data == key



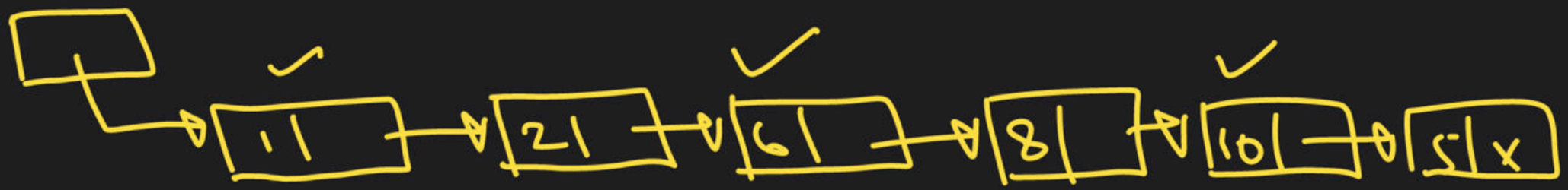

```
if (Ptr->data == Key)
{
    pf("Yes");
    return
}
Ptr = Ptr->Next;
```

repeat

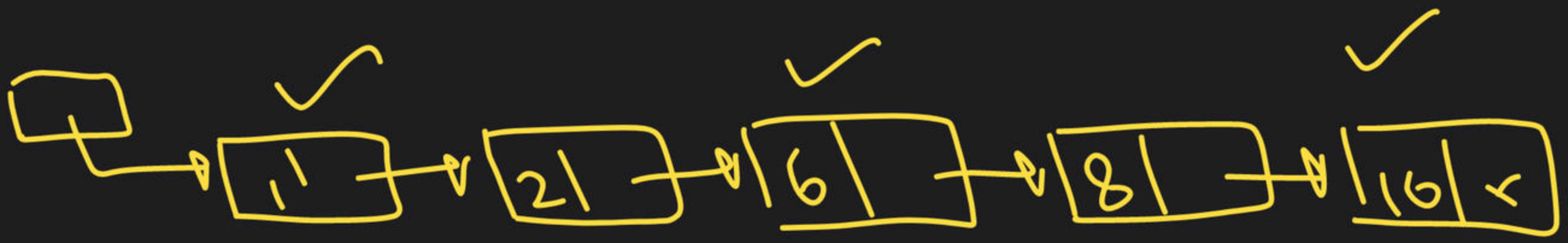



```
while(ptr != NULL)
{
if(ptr->data == key)
{
printf("Yes");
return;
}
ptr = ptr->Next;
}
printf("No");
```

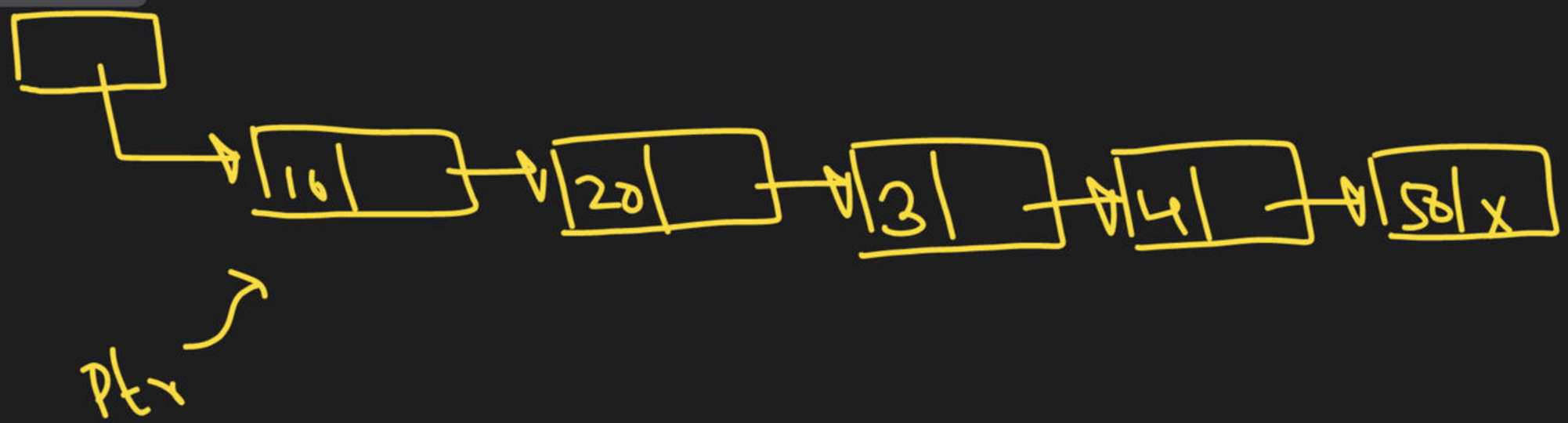
6. Given a linked list, print alternate node data.



o/p: 1 6 10

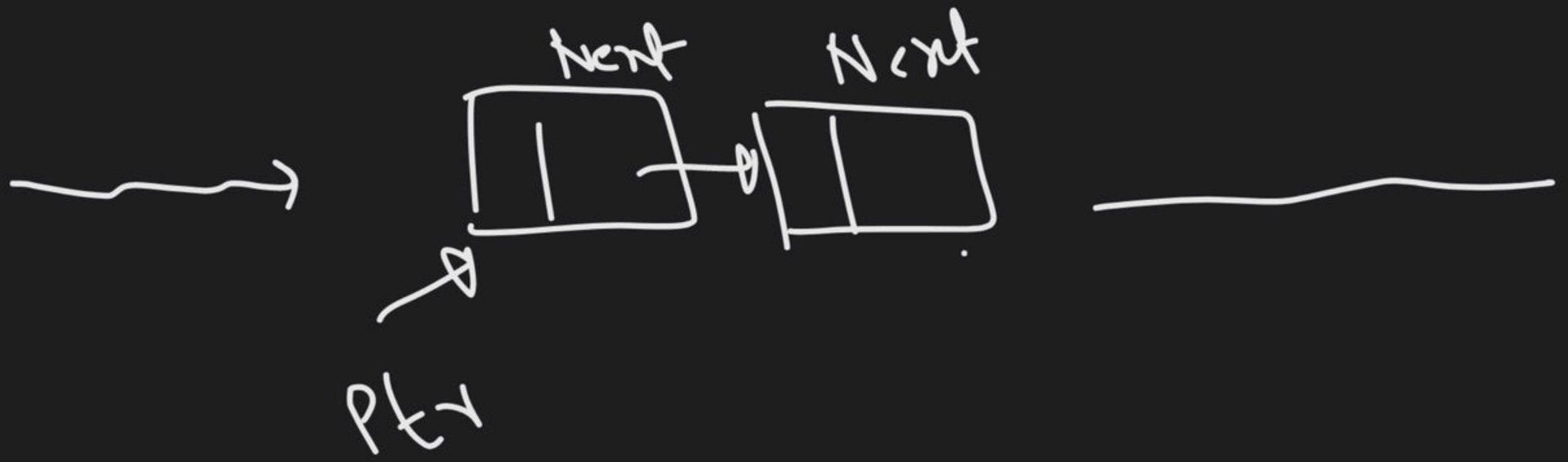


o/p: 1 6 10

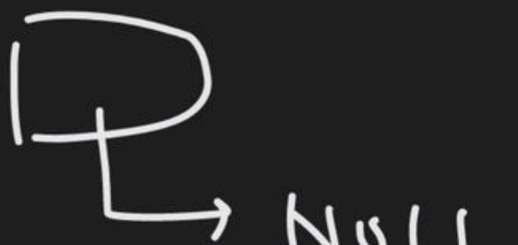


10

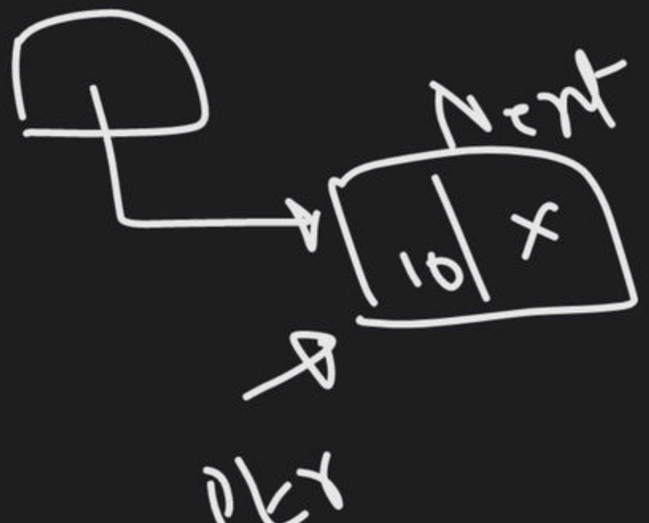
ptr = ptr → next → next ;



Case 1



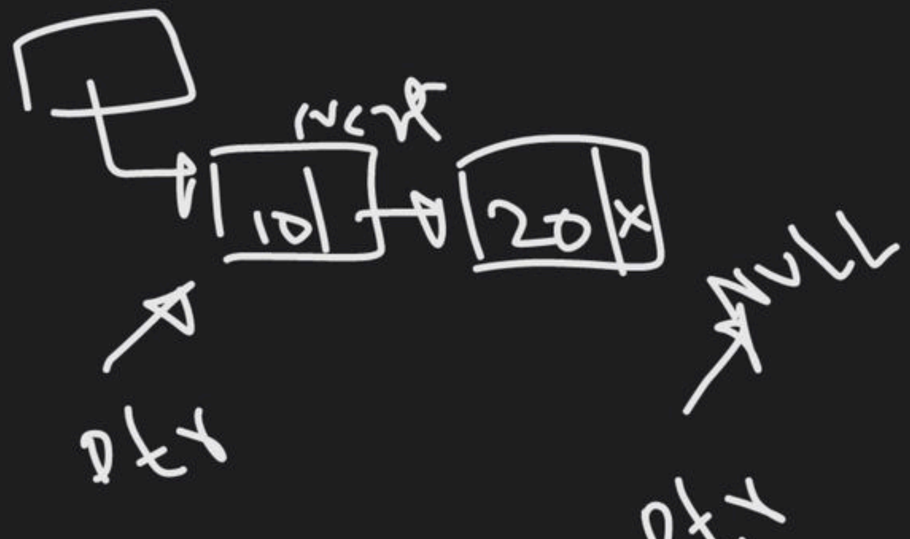
Case 2

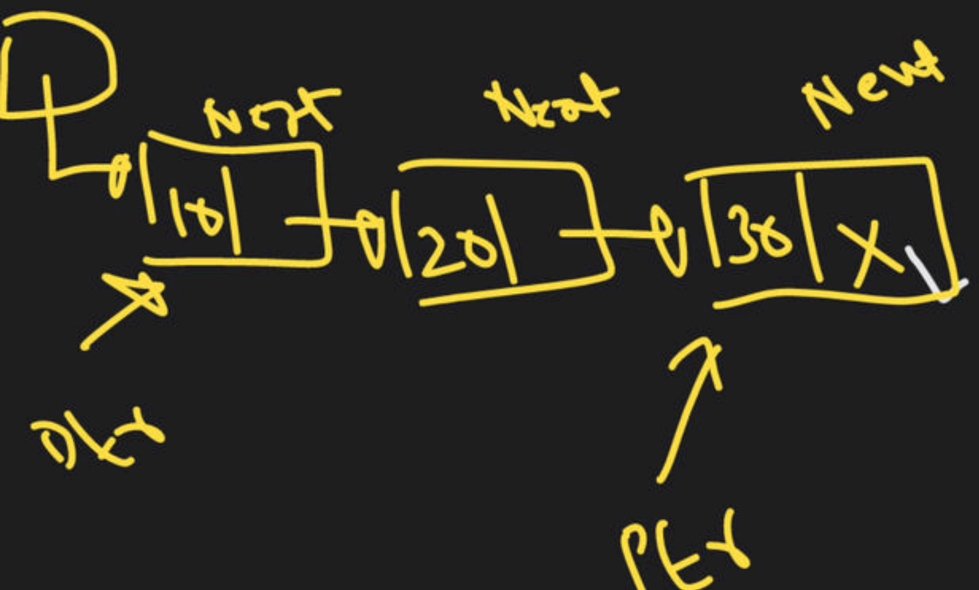



```
while (Ptr != NULL && Ptr->Next != NULL)
{
    pf("%d", Ptr->data);
    Ptr = Ptr->next->next;
}
```

10

```
while ( Ptr != NULL && Ptr -> Next !
                                = NULL)
{
    pf("%d", Ptr -> data);
    Ptr = Ptr -> next -> next ;
}
```






```
while (Ptr != NULL && Ptr->Next != NULL)
{
    pf("%d", Ptr->data);
    Ptr = Ptr->next->next;
}
```

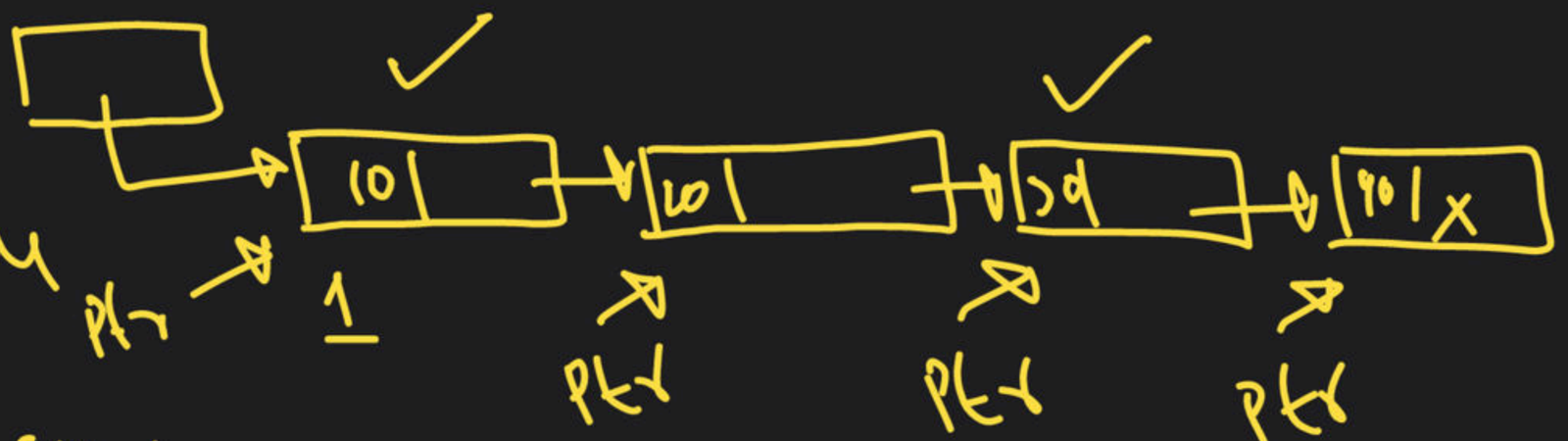
Odd node

Ptr 10

unacademy

```
while (ptr != NULL && ptr->next != NULL)
{
    pf("%d", ptr->data);
    ptr = ptr->next->next;
}
if (ptr == NULL)
    return;
if (ptr->next == NULL) pf("%d", ptr->data);
```

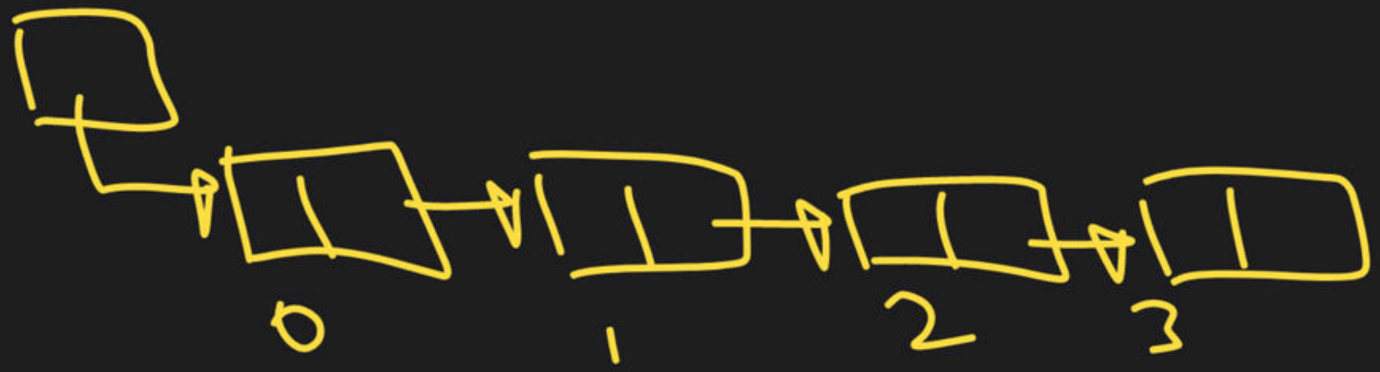
2nd way



count = 4 for 4th

```
int count=1; ptr=START
while(ptr!=NULL)
{
   if(count%2==1)
        printf("%d",ptr->data);
   ptr=ptr->next;
   count++;
}
```

```
int count = 0;

while(ptr != NULL)
 {

 if(count % 2 == 0)
        pf("%d", ptr->data);
  ptr = ptr->Next;
  count++;
 }
```



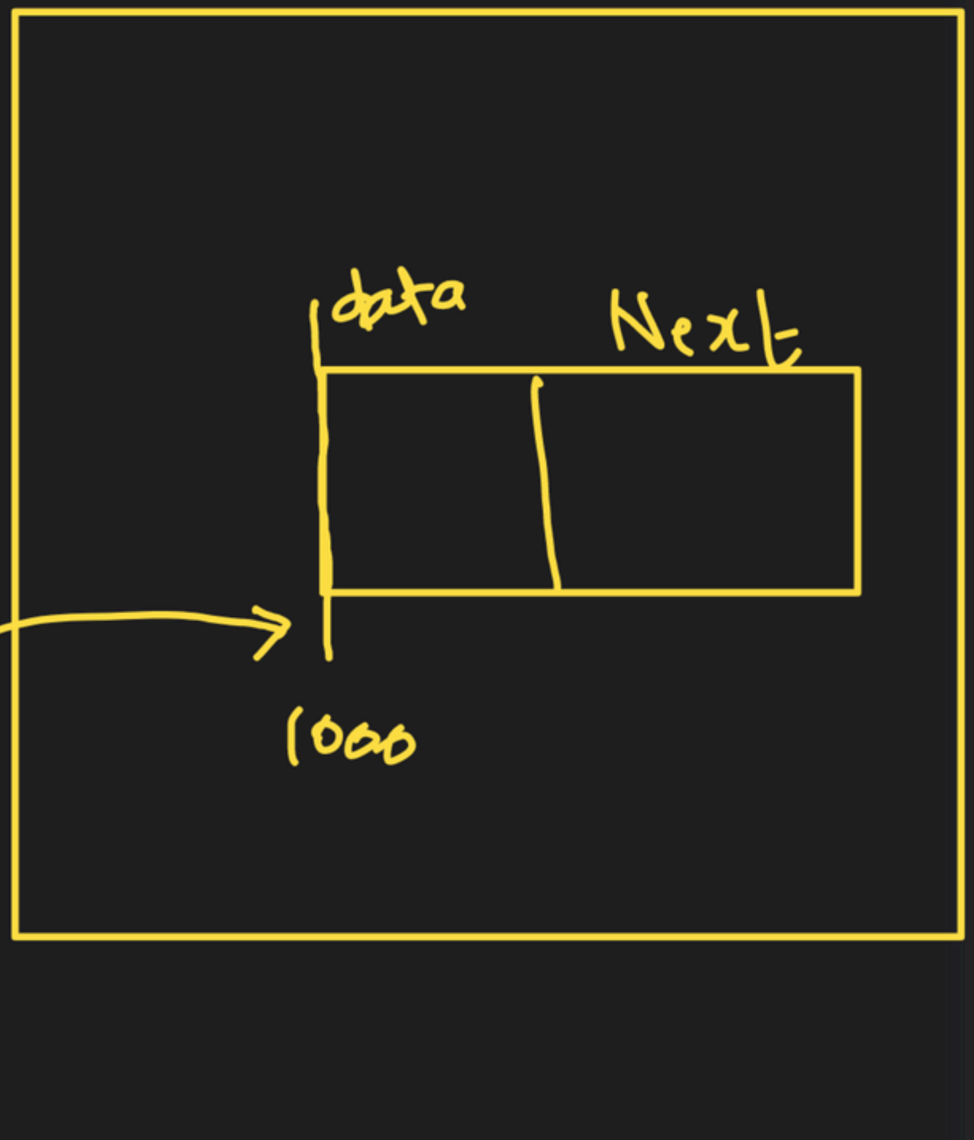
unacademy

Insertion

Element → Node

① Memory allocation

```
struct Node *temp;
temp = malloc(sizeof(struct Node));
```



① Memory allocation

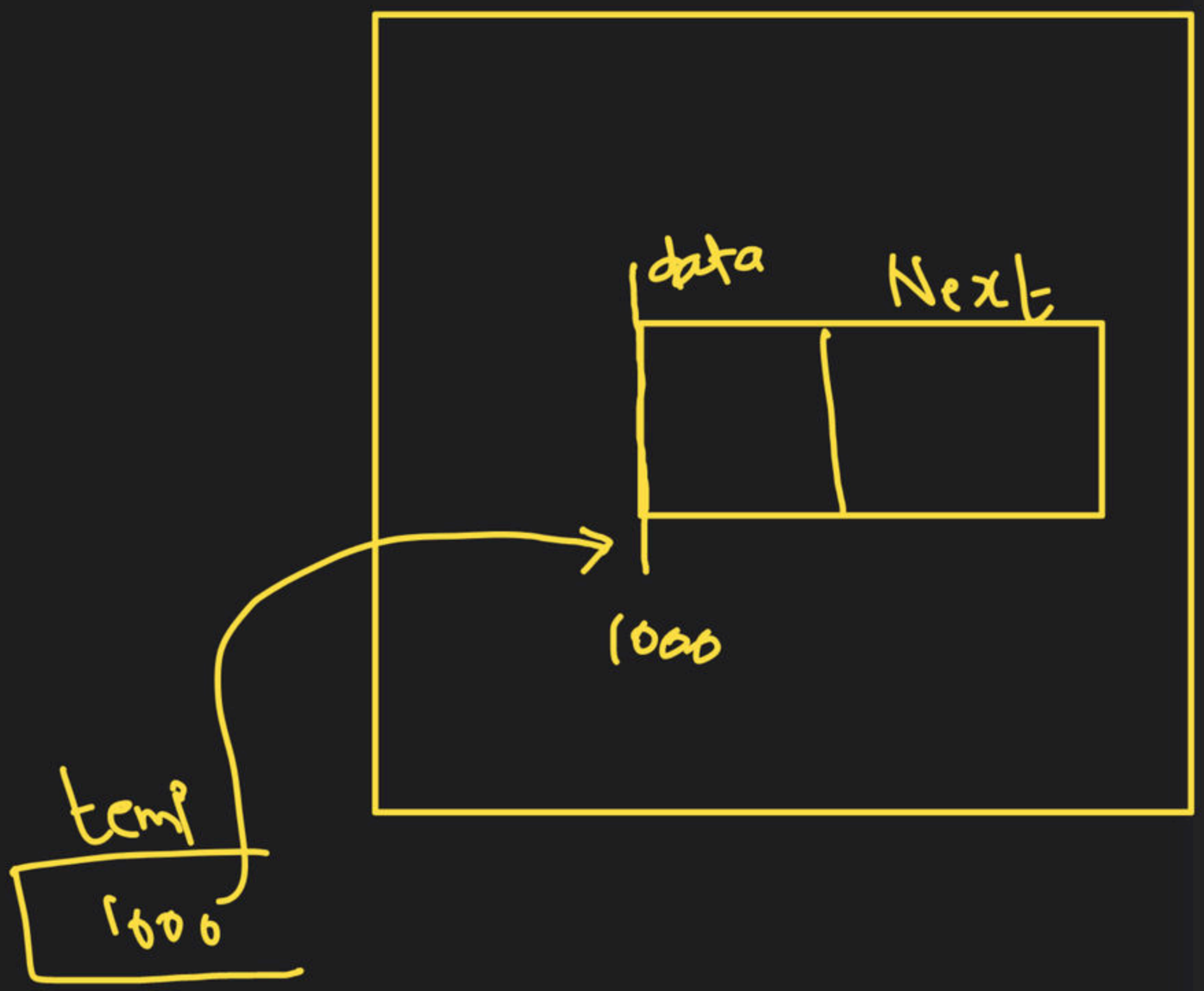
```
struct Node *temp;
temp = malloc(sizeof(Struct Node));
```

② Key insert

a) `temp -> data = key;`

b) Where to insert?

- begin
- End
- At position



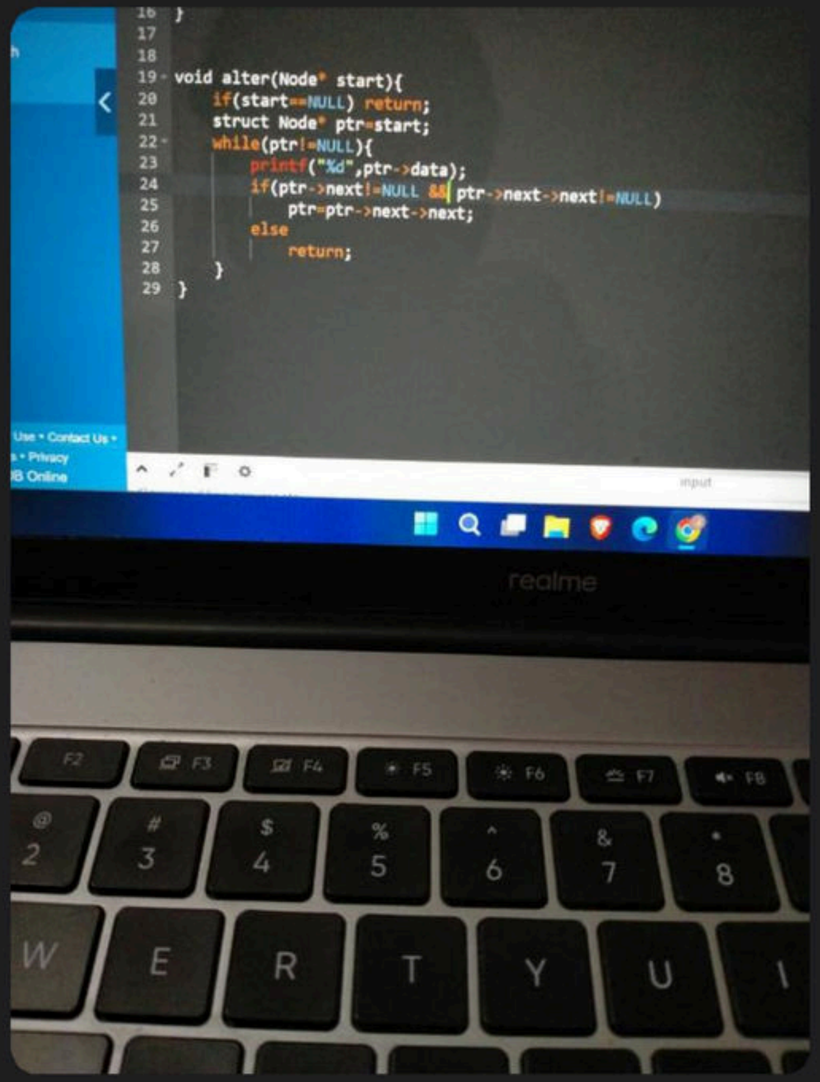
7) Given a L.L. and a key, insert the key at the beginning of LL.

$\simeq$

unacademy

▲ 1 • Asked by Jaiprakash

This?



unacademy

THANK YOU!

Here's to a cracking journey ahead!