**Placement Questions & Answers**

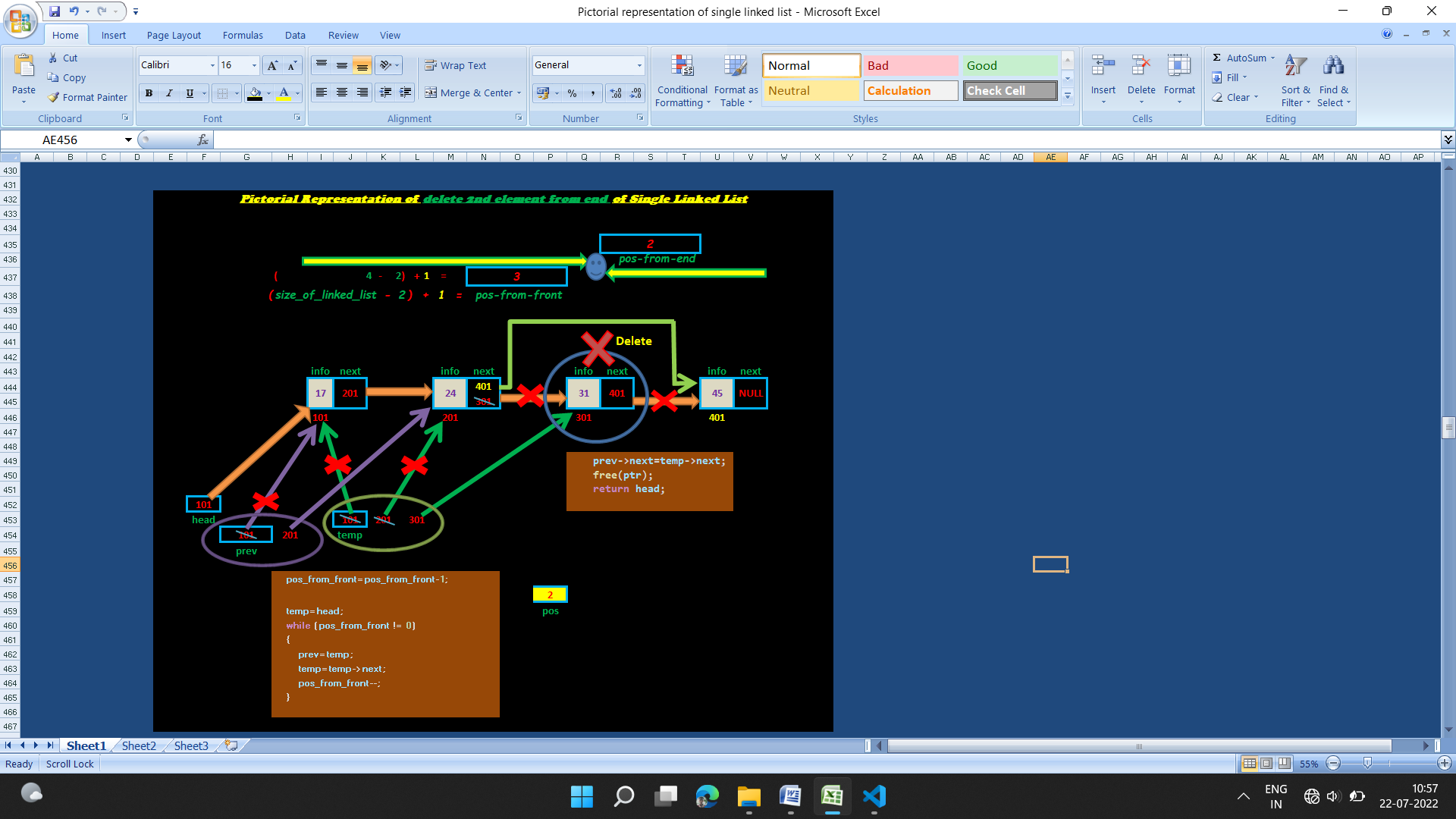
1. **Find 2nd node from last node and delete that node from single linked list.Considering that the head pointer is pointing to first node of the linked list.and size of the node is 5.**

**Ans:**

**If the position of the node from last is 2, then the position of that node from beginning will be 3**

**pos\_from\_end = ((size\_of\_linked\_list – pos\_from\_end) + 1)**

**pos\_from\_end = ((4 - 2 ) + 1) = 3**

****

**Code:**

//Find 2nd node from last node and delete that node from single linked list.

//Considering that the head pointer is pointing to first node of the linked list.

//and size of the node is 5.

#include<stdio.h>

#include<stdlib.h>

struct node\* create\_single\_linked\_list(struct node\*);

void print\_elements\_of\_single\_linked\_list(struct node\*);

struct node \* delete\_2nd\_node\_from\_end(struct node \*);

struct node  // Self-referentical structure

{

    int info;

    struct node\* next;

};

int main()

{

    struct node\* head;

    head=NULL;

    head=create\_single\_linked\_list(head);

    print\_elements\_of\_single\_linked\_list(head);

    head=delete\_2nd\_node\_from\_end(head);

    print\_elements\_of\_single\_linked\_list(head);

    return 0;

}

struct node \* delete\_2nd\_node\_from\_end(struct node \* head)

{

    //Size of linked list

    int count,size\_of\_linked\_list;

    struct node \* ptr;

    count=0;

    ptr=head;

    while (ptr != NULL)

    {

        count++;

        ptr=ptr->next;

    }

    size\_of\_linked\_list=count;

    printf("Size of linked list= %d\n",size\_of\_linked\_list);

    //Calculating position of node from front

    // If the position of the node from last is 2

    //then the position of that node from front will be

    //(size\_\_of\_linked\_list - 2) +1)

    //((      5             - 2) +1)

    int pos\_from\_end,pos\_from\_front;

    pos\_from\_end=2;

    pos\_from\_front=((size\_of\_linked\_list - pos\_from\_end)+1);

    struct node \* temp,\*prev;

    pos\_from\_front=pos\_from\_front-1; //for loop execution

    temp=head;

    while (pos\_from\_front != 0)

    {

        prev=temp;

        temp=temp->next;

        pos\_from\_front--;

    }

    prev->next=temp->next;

    free(temp);

    temp=NULL;

    prev=NULL;

    return head;

}

struct node\* create\_single\_linked\_list(struct node\* start)

{

    int n;

    int count;

    char ans;

    struct node\* temp;

    temp=malloc(sizeof(struct node));

    printf("Enter %d item for linked list:",1);

    scanf("%d",&temp->info);

    start=temp;

    count=1;

    while (count!=5)

    {

        temp->next=malloc(sizeof(struct node));

        temp=temp->next;

        printf("Enter %d item for linked list:",count+1);

        scanf("%d",&temp->info);

        count++;

    }

    temp->next=NULL;

    return start;

}

void print\_elements\_of\_single\_linked\_list(struct node\* head)

{

    while (head!=NULL)

    {

        printf("%d ",head->info);

        head=head->next;

    }

    printf("\n");

}

**Output:**

**Enter 1 item for linked list: 34**

**Enter 2 item for linked list: 46**

**Enter 3 item for linked list: 67**

**Enter 4 item for linked list: 43**

**Enter 5 item for linked list: 56**

**34 46 67 43 56**

**Size of linked list= 5**

**34 46 67 56**

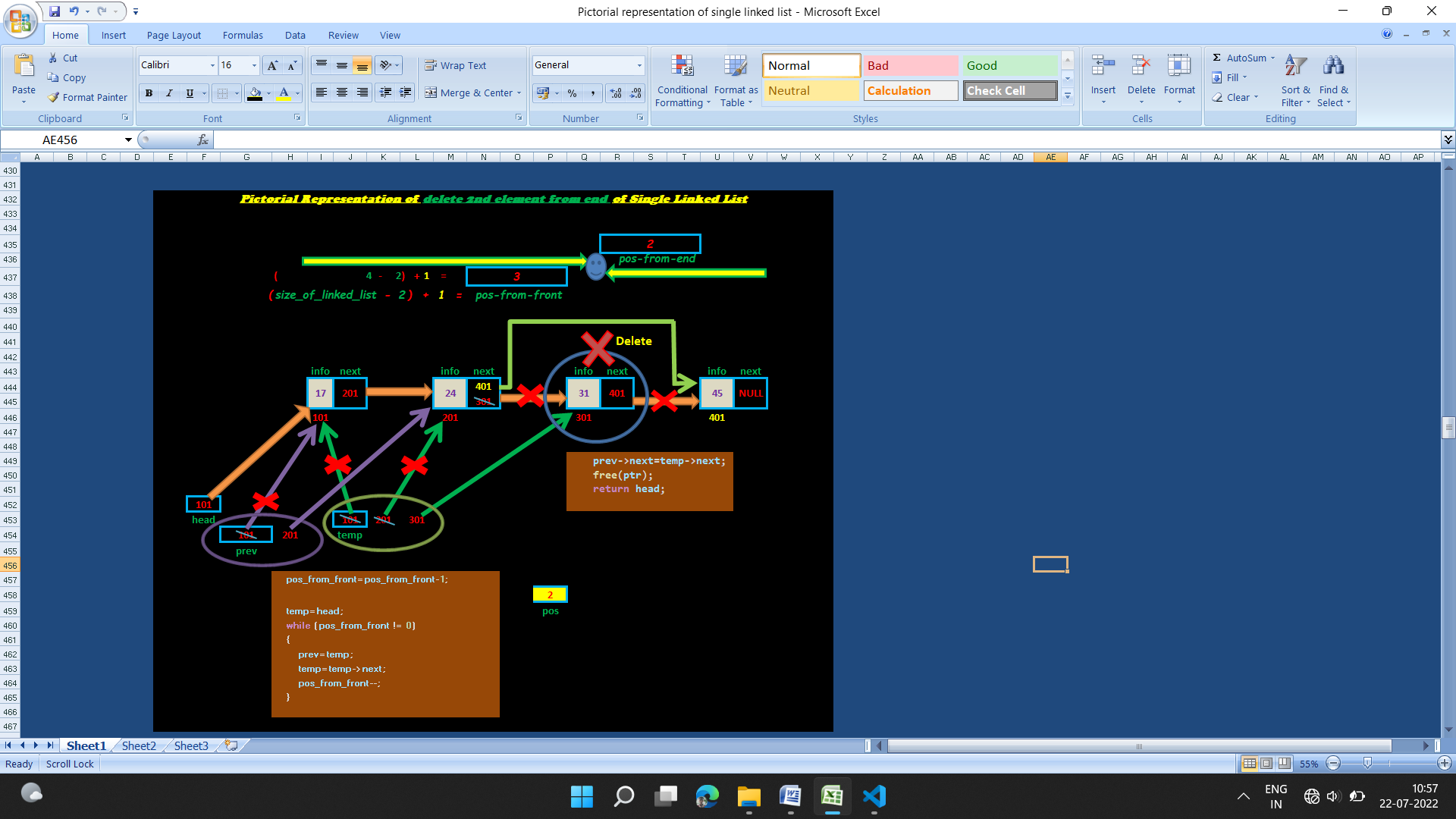
**2)Find 2nd node from last node and delete that node from single linked list.Considering that the head pointer is pointing to first node of the linked list.and size of the node is 5.**

**Ans:**

**If the position of the node from last is 2, then the position of that node from beginning will be 3**

**pos\_from\_end = ((size\_of\_linked\_list – pos\_from\_end) + 1)**

**pos\_from\_end = ((4 - 2 ) + 1) = 3**

****

**Code:**

//Find 2nd node from last node and delete that node from single linked list.

//Considering that the head pointer is pointing to first node of the linked list.

//and size of the node is 5.

#include<stdio.h>

#include<stdlib.h>

struct node\* create\_single\_linked\_list(struct node\*);

void print\_elements\_of\_single\_linked\_list(struct node\*);

struct node \* delete\_2nd\_node\_from\_end(struct node \*);

struct node  // Self-referentical structure

{

    int info;

    struct node\* next;

};

int main()

{

    struct node\* head;

    head=NULL;

    head=create\_single\_linked\_list(head);

    print\_elements\_of\_single\_linked\_list(head);

    head=delete\_2nd\_node\_from\_end(head);

    print\_elements\_of\_single\_linked\_list(head);

    return 0;

}

struct node \* delete\_2nd\_node\_from\_end(struct node \* head)

{

    //Size of linked list

    int count,size\_of\_linked\_list;

    struct node \* ptr;

    count=0;

    ptr=head;

    while (ptr != NULL)

    {

        count++;

        ptr=ptr->next;

    }

    size\_of\_linked\_list=count;

    printf("Size of linked list= %d\n",size\_of\_linked\_list);

    //Calculating position of node from front

    // If the position of the node from last is 2

    //then the position of that node from front will be

    //(size\_\_of\_linked\_list - 2) +1)

    //((      5             - 2) +1)

    int pos\_from\_end,pos\_from\_front;

    pos\_from\_end=2;

    pos\_from\_front=((size\_of\_linked\_list - pos\_from\_end)+1);

    struct node \* temp,\*prev;

    pos\_from\_front=pos\_from\_front-1;

    temp=head;

    while (pos\_from\_front != 0)

    {

        prev=temp;

        temp=temp->next;

        pos\_from\_front--;

    }

    prev->next=temp->next;

    free(temp);

    temp=NULL;

    prev=NULL;

    return head;

}

struct node\* create\_single\_linked\_list(struct node\* start)

{

    int n;

    int count;

    char ans;

    struct node\* temp;

    temp=malloc(sizeof(struct node));

    printf("Enter %d item for linked list:",1);

    scanf("%d",&temp->info);

    start=temp;

    count=1;

    while (count!=5)

    {

        temp->next=malloc(sizeof(struct node));

        temp=temp->next;

        printf("Enter %d item for linked list:",count+1);

        scanf("%d",&temp->info);

        count++;

    }

    temp->next=NULL;

    return start;

}

void print\_elements\_of\_single\_linked\_list(struct node\* head)

{

    while (head!=NULL)

    {

        printf("%d ",head->info);

        head=head->next;

    }

    printf("\n");

}