10/11/2025

4) Write a program to implement singly linked list with the following operation

a) create a linked list

b) Insertion of a node at → first position
→ any position
→ end of the list

c) Display the contents of linked list

Pseudo

```
struct node {
    int data;
    struct node *next;
};
struct node *head=NULL
Newnode->data=data;
Newnode->next=NULL;
if(head==NULL){
    head=newnode=temp
}
else {
    temp->next=newnode;
    temp=newnode
}
```

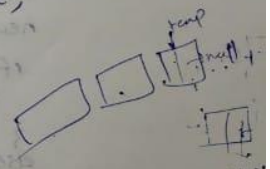
Insertion

a) void Insertatbeginning(int data) {

```
struct node *newnode=(struct node*) malloc (sizeof(struct node));
newnode->data=data;
newnode->next=newNod head;
head=newnode
Printf
}
```

```
void InsertAtEnd(int data){
    newnode->data=data;
    newnode->next=NULL;
    if(head==NULL){
        head=newnode;
    }
    else{
        Struct Node* temp=head;
        while(temp->next!=NULL){
            temp=temp->next;
        }
        temp->next=newnode;
}
```

```
Void InsertAtposition(int data){
    newnode->data=data;
for(i=1; i=pos-1 && temp->next!=NULL ; i++){
        temp=temp->next;
    if(temp==NULL){
        free(newData);
    } else{
        newnode->next=temp->next;
        temp->next=new node;
    }
```

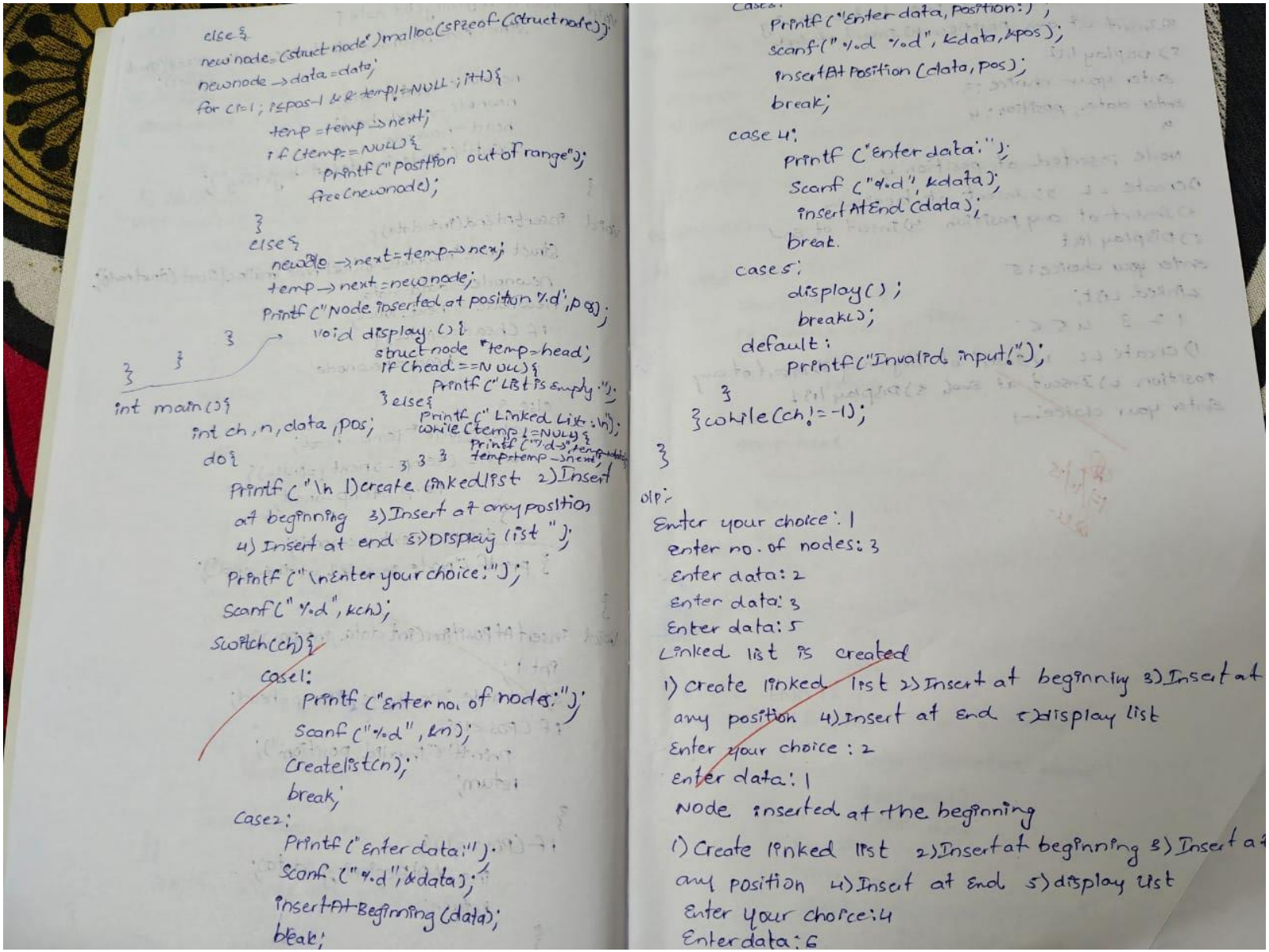



```
Void Display(){
    for while(temp!=NULL)
    {temp=head;
    printf("%d",temp->data);
    temp=temp->next;
}
```

code:-

```
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node *next;
}; struct node *head=NULL;
void createlist(int n){
    struct node *newnode, *temp;
    int data, i;
    if (n<0) {
        Printf("n should be greater than 0");
        return;
    }
    for (i=1; i<=n; i++){
        newnode=(struct node*)malloc(sizeof(struct node));
        if (newnode==NULL) {
            Printf("Memory allocation failed.");
            return;
        }
        Printf("Enter data: ");
        scanf("%d", &data)
        newnode->data=data;
        newnode->next=NULL;
        if (head==NULL) {
            head=newnode;
        }
        else {
            temp->next=newnode;
        }
        temp->next=newnode;
    }}
    Printf("Linked list is created.");
}
```

```
void insertAtBeginning(int data) {
    struct node *newnode=(struct node*) malloc(sizeof(struct node));
    newnode->data=data;
    newnode->next=head;
    head=newnode;
    Printf("Node inserted at beginning");
}
void insertAtEnd(int data){
    Struct node *newnode=(struct node*)malloc(sizeof(struct node));
    newnode->data=data;
    newnode->next=NULL;
    if (head==NULL) {
        head=newnode;
    }
    else {
        struct node *temp=head;
        while (temp->next!=NULL){
            temp=temp->next;
        }
        temp->next=newnode;
    } printf("Node inserted at the end");
}
void insertAtPosition(int data, int pos){
    int i;
    struct node *newnode, *temp=head;
    if (pos<1){
        Printf("Invalid position:");
        return;
    }
    if (pos==1) {
        insertAtBeginning(data);
        return;
    }
```



```
else{
newnode=(struct node*)malloc(sizeof(struct node));
newnode->data=data;
for(i=1;i<pos-1 && temp!=NULL;i++){
    temp=temp->next;
    if(temp==NULL){
        printf("position out of range");
        free(newnode);
    }
    else{
        newnode->next=temp->next;
        temp->next=newnode;
        Printf("Node inserted at position %d",pos);
    }
}
void display(){
    struct node *temp=head;
    if(head==NULL){
        printf("List is empty.");
    }else{
        Printf("Linked List:\n");
        while(temp!=NULL){
            Printf("%d->",temp->data);
            temp=temp->next;
        }
    }
}
int main(){
    int ch,n,data,pos;
    do{
        Printf("\n 1)create linkedlist 2)Insert at beginning 3)Insert at any position 4)Insert at end 5)Display list");
        Printf("\nenter your choice:");
        Scanf("%d",&ch);
        Switch(ch){
            case1:
                Printf("Enter no. of nodes:");
                Scanf("%d",&n);
                Createlist(n);
                break;
            case2:
                Printf("Enter data:");
                Scanf("%d",&data);
                InsertAtBeginning(data);
                break;
            case3:
                Printf("Enter data, position:");
                Scanf("%d %d",&data,&pos);
                InsertAtPosition(data,pos);
                break;
            case4:
                Printf("Enter data:");
                Scanf("%d",&data);
                InsertAtEnd(data);
                break;
            case5:
                display();
                break;
            default:
                Printf("Invalid input!");
        }
    }while(ch!=-1);
}
```

```
o/p:
Enter your choice: 1
enter no. of nodes: 3
Enter data: 2
Enter data: 3
Enter data: 5
Linked list is created
1) Create linked list 2) Insert at beginning 3) Insert at any position 4) Insert at end 5) display list
Enter your choice: 2
enter data: 1
Node inserted at the beginning
1) Create linked list 2) Insert at beginning 3) Insert at any position 4) Insert at end 5) display list
Enter your choice: 4
Enter data: 6
```

1) Create linked list 2) Insert at beginning
3) Insert at any position 4) Insert at end
5) Display list
Enter your choice: 3
Enter data, position: 4
4
Node inserted at position 4.
1) create LL 2) Insert at beginning
3) Insert at any position 4) Insert at end
5) Display list
enter your choice: 5
Linked List;
1 2 3 4 5 6
1) Create LL 2) Insert at beginning 3) Insert at any position 4) Insert at end 5) Display list
Enter your choice: 1

17/11/25