

Q1.

```
#include <stdio.h>
#include <stdlib.h>

struct Node{
    int data;
    struct Node * next;
};

struct Node *showLinkList(struct Node *ptr)
{
    if (ptr == NULL)
    {
        printf("\n\nLinked list is Empty.");
    }
    printf("\nLink list contains : ");
    while (ptr != NULL)
    {
        printf("%d ", ptr->data);
        ptr = ptr->next;
    }
    printf("-> NULL (End of Linked List)");
}

struct Node *insertEnd(struct Node * head){
    struct Node * new = (struct Node*)malloc(sizeof(struct Node));
    printf("Enter value : ");
    scanf("%d", &new->data);

    if(head==NULL){
        head=new;
        head->next=NULL;
        return head;
    }
    struct Node * temp = head;
    while(temp->next!=NULL){
        temp=temp->next;
    }
    new->next=NULL;
    temp->next=new;
    return head;
}

struct Node *insertAtSpecificPosition(struct Node *head){
    struct Node * new = (struct Node*)malloc(sizeof(struct Node));
```

```

printf("\nEnter value : ");
scanf("%d", &new->data);

int i = 0, pos=0;

printf("Enter position : ");
scanf("%d", &pos);printf("\n");

struct Node * temp = head;

while (i!=pos-1)
{
    temp=temp->next;
    i++;
}

new->next=temp->next;
temp->next=new;

return head;
}

int main(){
    struct Node *head;
    head = NULL;

    for (int x = 0; x < 5; x++){
        head=insertEnd(head);
    }

    printf("\nAfter Operation\n");

    showLinkedList(head);

    head = insertAtSpecificPosition(head);

    showLinkedList(head);

    return 0;
}

```

Q2.

```
#include <stdio.h>
#include <stdlib.h>

struct Node
{
    int data;
    struct Node *next;
};

struct Node *showLinkList(struct Node *ptr)
{
    if (ptr == NULL){
        printf("\n\nLinked list is Empty.");
    }
    printf("\nLink list contains : ");
    while (ptr != NULL){
        printf("%d ", ptr->data);
        ptr = ptr->next;
    }
    printf("-> NULL (End of Linked List)");
}

struct Node *insertEnd(struct Node * head){
    struct Node * new = (struct Node*)malloc(sizeof(struct Node));
    printf("Enter value : ");
    scanf("%d", &new->data);

    if(head==NULL){
        head=new;
        head->next=NULL;
        return head;
    }
    struct Node * temp = head;
    while(temp->next!=NULL){
        temp=temp->next;
    }
    new->next=NULL;
    temp->next=new;
    return head;
}

struct Node * search(struct Node * ptr, int element){
    if(ptr==NULL){
```

```

        printf("\nNo Element Present.");
    }
    else{
        int pos=0, cnt=0;
        struct Node * temp=ptr;
        while (temp!=NULL)
        {
            if(temp->data==element){
                printf("\nElement found at position %d and value :-
%d", pos+1,element);
                cnt++;
                exit(0);
            }
            pos++;
            temp=temp->next;
        }
        if(cnt==0){
            printf("\nValue not found.");
        }
    }
}

int main(){

    struct Node *head=NULL;
    for (int x = 0; x < 5; x++){
        head=insertEnd(head);
    }

    showLinkList(head);

    search(head, 30);

    return 0;
}

```

Q3.

```
#include <stdio.h>
#include <stdlib.h>

struct Node
{
    int data;
    struct Node *next;
};

int sum=0;

struct Node *showLinkList(struct Node *ptr)
{
    if (ptr == NULL){
        printf("\n\nLinked list is Empty.");
    }
    printf("\nLink list contains : ");
    while (ptr != NULL){
        printf("%d ", ptr->data);
        ptr = ptr->next;
    }
    printf("-> NULL (End of Linked List)");
}

struct Node *insertEnd(struct Node * head){
    struct Node * new = (struct Node*)malloc(sizeof(struct Node));
    printf("Enter value : ");
    scanf("%d", &new->data);

    if(head==NULL){
        head=new;
        head->next=NULL;
        return head;
    }
    struct Node * temp = head;
    while(temp->next!=NULL){
        temp=temp->next;
    }
    new->next=NULL;
    temp->next=new;
    return head;
}

int sumUp(struct Node * ptr){
```

```
while (ptr!=NULL)
{
    sum+=ptr->data;
    ptr=ptr->next;
}
return sum;
}

int main(){

    struct Node *head=NULL;
    for (int x = 0; x < 5; x++){
        head=insertEnd(head);
    }

    showLinkList(head);

    printf("\n\nSum = %d", sumUp(head));

    return 0;
}
```