

```
//operations on a doubly linked list  
// inserting at the beginning (insertion)
```

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

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

```
struct node
```

```
{  
    int data;  
    struct node* next;  
    struct node* prev;  
};
```

```
struct node* insertbeginning(int item, struct node* head)
```

```
{  
    struct node *ptr = (struct node *)malloc(sizeof(struct node));  
    if(head==0)  
    {  
        ptr->next = 0;  
        ptr->prev=0;  
        ptr->data=item;  
        head=ptr;  
    }  
    else{  
        ptr->data=item;  
        ptr->prev=0;  
        ptr->next = head;  
        head->prev=ptr;  
        head=ptr;  
    }  
    return head;
```

```
}
```

```
struct node* insertlast(int item, struct node* head)
```

```
{
```

```
    struct node *ptr = (struct node *) malloc(sizeof(struct node));
```

```
    struct node *temp;
```

```
    ptr->data=item;
```

```
    if(head == 0)
```

```
    {
```

```
        ptr->next = 0;
```

```
        ptr->prev = 0;
```

```
        head = ptr;
```

```
    }
```

```
    else
```

```
    {
```

```
        temp = head;
```

```
        while(temp->next!=0)
```

```
        {
```

```
            temp = temp->next;
```

```
        }
```

```
        temp->next = ptr;
```

```
        ptr->prev=temp;
```

```
        ptr->next = 0;
```

```
    }
```

```
    return head;
```

```
}
```

```
void print(struct node* head)
```

```
{
```

```
    struct node* ptr = head;
```

```
    int i=1;
```

```
while(ptr!=0)
{
    printf("\nElement in %d node is %d",i,ptr->data);
    i++;
    ptr=ptr->next;
}
}
```

```
void searchNode(struct node* head, int item)
{
    struct node* ptr = head;
    int found=0,i=1;
    while(ptr!=0)
    {
        if(ptr->data==item)
        {
            printf("\nData found at %d node and at %p location",i,ptr);
            found = 1;
            break;
        }
        i++;
        ptr=ptr->next;
    }
    if(found==0){
        printf("\nGiven data not in linked list");
    }
}
```

```
int main()
{
    int n;
```

```

struct node* head = (struct node *)malloc(sizeof(struct node));

head->data =3;

head->next=0;

head->prev=0;

while(1)

{

    printf("\n1.Insert at beginning\n2.Insert at end\n3.Search for an item\n4.Print linked list
elements\n5.Exit\n");

    printf("Enter a choice:");

    scanf("%d",&n);

    if(n==1)

    {

        int data;

        printf("\nEnter the data\n");

        scanf("%d",&data);

        head = insertbeginning(data,head);

    }

    else if(n==2)

    {

        int data;

        printf("\nEnter the data\n");

        scanf("%d",&data);

        head =insertlast(data,head);

    }

    else if(n==4)

    {

        print(head);

    }

    else if(n==3)

    {

        int data;

```

```
        printf("\nEnter the data\n");
        scanf("%d",&data);
        searchNode(head,data);
    }
    else if(n==5)
    {
        exit(0);
    }
    else{
        printf("\nInvalid input\n");
    }
}
return 0;
}
```

OUTPUT:

```
1.Insert at beginning
2.Insert at end
3.Search for an item
4.Print linked list elements
5.Exit
Enter a choice:1

Enter the data
20

1.Insert at beginning
2.Insert at end
3.Search for an item
4.Print linked list elements
5.Exit
Enter a choice:2

Enter the data
50

1.Insert at beginning
2.Insert at end
3.Search for an item
4.Print linked list elements
5.Exit
Enter a choice:3

Enter the data
20

Data found at 1 node and at 0000000001C1450 location
1.Insert at beginning
2.Insert at end
3.Search for an item
4.Print linked list elements
5.Exit
Enter a choice:3

Enter the data
55

Given data not in linked list
1.Insert at beginning
2.Insert at end
3.Search for an item
4.Print linked list elements
5.Exit
Enter a choice:4

Element in 1 node is 20
Element in 2 node is 10
Element in 3 node is 50
1.Insert at beginning
2.Insert at end
3.Search for an item
4.Print linked list elements
5.Exit
Enter a choice:5

Process returned 0 (0x0)   execution time : 54.666 s
Press any key to continue.
```