```
}


#include<stdio.h>

#include <unistd.h>


#include<stdlib.h>

struct node

{

int value;

struct node *next;

};

struct node *head;


void init()

{

        head = NULL ;

}


void insertfirst(int element)

{

        struct node *New;

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

        New->value=element;

        New->next=NULL;

        New->next=head;

        head=New;

}


void print()
```

```
{
    if(head==NULL)
    {
            printf("list is empty\n");
            return;
    }
    struct node *cur=head;
    int count;
    count=0;
    while(cur!=NULL)
    {
        printf("%d->",cur->value);
        count++;
        cur=cur->next;
    }
    printf("NULL\n");
    printf("number of nodes %d\n",count);
}

void deleteitem(int ele)
{
    if(head==NULL)
        printf("list is empty and nothing to delete\n");
    struct node* cur=head;
    struct node* prev=NULL;
    while(cur->value!=ele)
    {
```

```
                prev=cur;
                cur=cur->next;
        }
        if(prev!=NULL)
                prev->next=cur->next;
        free(cur);
}
int searchitem(int ele)
{
        struct node* temp ;
        temp = head;
        while (temp != 0)
        {
                if (temp->value == ele)
                        return 1 ;
                temp = temp->next;
        }

        return 0 ;
}
void insertlast(int ele)
{
        struct node *New, *temp;
        New = (struct node*)malloc(sizeof(struct node));
        if(New== NULL)
        {
                printf("Unable to allocate memory.");
        return;
    }
    else
    {
        New->value = ele;
        New->next = NULL;
        temp = head;
        while(temp->next != NULL)
            temp = temp->next;
        temp->next = New;
        printf("DATA INSERTED SUCCESSFULLY\n");
    }
}
void deletelast()
{
    if(head==NULL)
        printf("list is empty and nothing to delete\n");
    struct node* cur=head;
    struct node* prev=NULL;
    while(cur->next!=NULL)
    {
        prev=cur;
        cur=cur->next;
    }
    if(prev->next!=NULL)
        prev->next=NULL;
    free(cur);
```

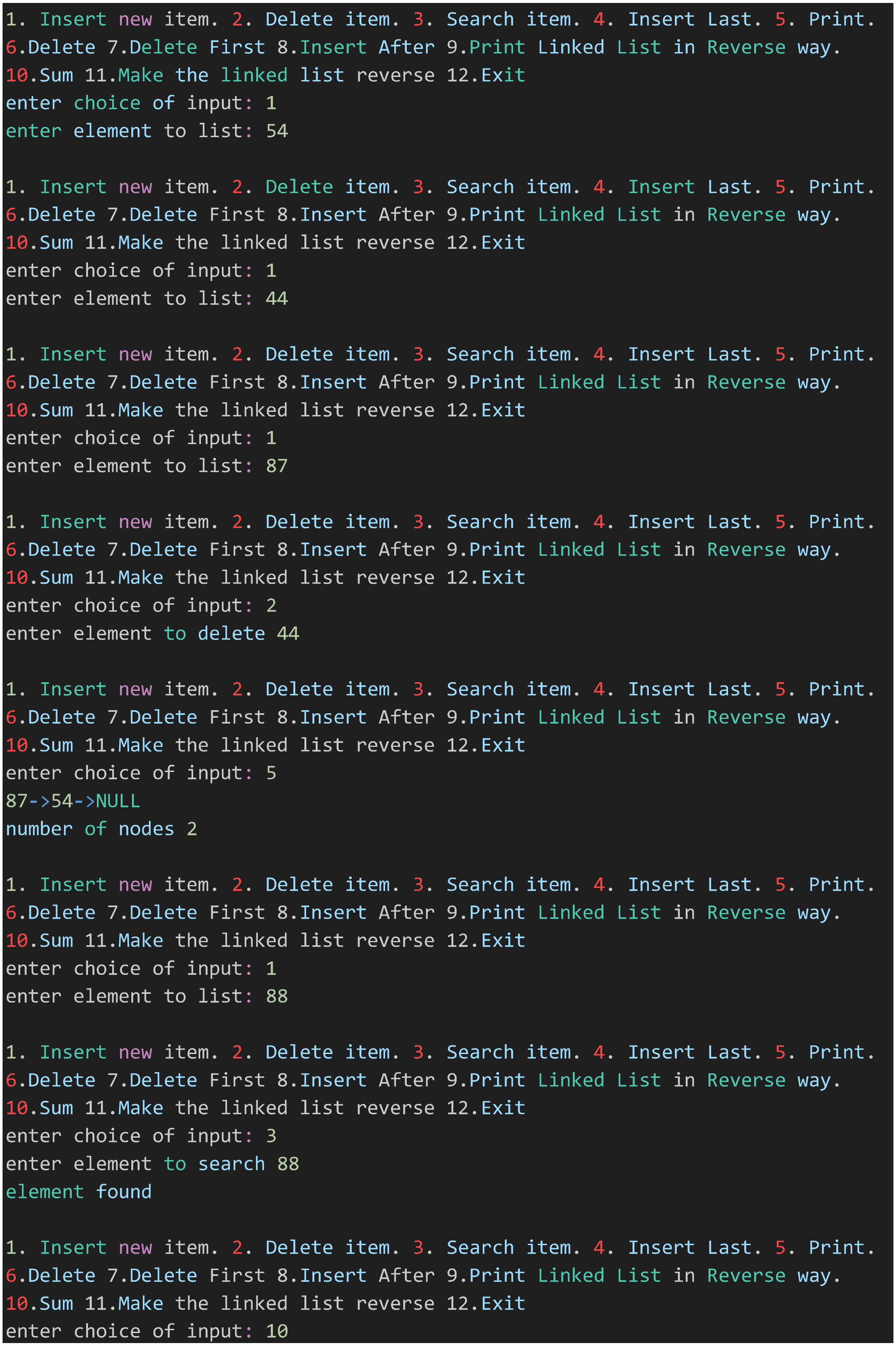
```
}
void deletefirst()
{
    struct node* cur;
    if(head==NULL)
        printf("list is empty and nothing to delete\n");
    cur=head;
    head=head->next;
    free(cur);
}
void insertafter(int elem, int num)
{
    struct node* New;
    New=(struct node*)malloc(sizeof(struct node));
    New->value=elem;
    New->next=NULL;
    struct node* prev=head;
    while(prev->value!=num)
        prev=prev->next;
    New->next=prev->next;
    prev->next=New;
}
void printReverse(struct node* head)
{
    if (head == NULL)
        return;
    printReverse(head->next);
```

```
        printf("%d->", head->value);
}
void reverselist()
{
        struct node* prev=NULL;
        struct node* cur=head;
        struct node* nxt;
        while(cur!=NULL)
        {
                nxt=cur->next;
                cur->next=prev;
                prev=cur;
                cur=nxt;
        }
        head=prev;
}
void sum()
{
        int s;
        struct node *cur=head;
        s=0;
        while(cur!=NULL)
        {
                s+=cur->value;
                cur=cur->next;
        }
        printf("Sum of elements is %d\n",s);
```

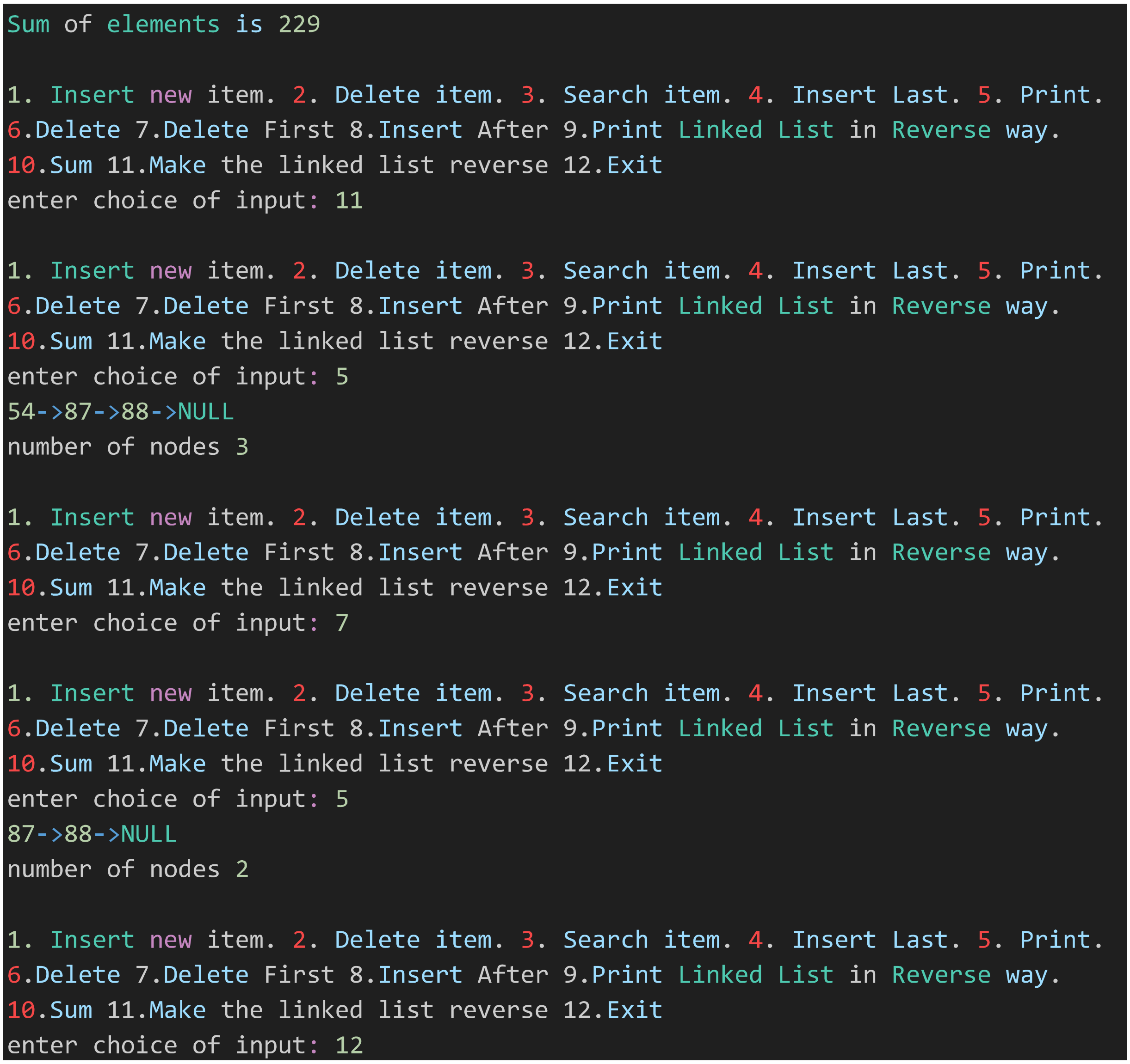
```
}

int main()

{

        init();

        int ch,element;

        while(1)

        {


                printf("\n1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way. 10.Sum 11.Make the linked
list reverse 12.Exit\n");

                printf("enter choice of input: ");

                scanf("%d",&ch);

                if(ch==1)

                {

                        printf("enter element to list: ");

                        scanf("%d",&element);

                        insertfirst(element);


                }

                else if(ch==2)

                {

                        int de;

                        printf("enter element to delete ");

                        scanf("%d",&de);

                        deleteitem(de);


                }
```

```
}

else if(ch==3)

{

        int se,reply;

        printf("enter element to search ");

        scanf("%d",&se);

        reply=searchitem(se);

        if(reply==1)

                printf("element found\n");

        else

                printf("element not found\n");


}

else if(ch==4)

{

        int le;

        printf("enter element to insert at last ");

        scanf("%d",&le);

        insertlast(le);


}

else if(ch==5)

{

        print();


}

else if(ch==6)

{
        deletelast();

}
else if(ch==7)
{
        deletefirst();

}
else if(ch==8)
{
        int ie,p;
        printf("enter element to insert: ");
        scanf("%d",&ie);
        printf("enter after which element to insert: ");
        scanf("%d",&p);
        insertafter(ie,p);

}
else if(ch==9)
{
        printReverse(head);

}
else if(ch==10)
{
        sum();
        }
        else if(ch==11)
        {
            reverselist();

        }
        else if(ch==12)
            return 0;
        else
        {
            printf("INCORRECT INPUT");
            return(0);
        }


    }


    return 0;

}
```

OUTPUT



```
1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way.
10.Sum 11.Make the linked list reverse 12.Exit
enter choice of input: 1
enter element to list: 54

1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way.
10.Sum 11.Make the linked list reverse 12.Exit
enter choice of input: 1
enter element to list: 44

1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way.
10.Sum 11.Make the linked list reverse 12.Exit
enter choice of input: 1
enter element to list: 87

1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way.
10.Sum 11.Make the linked list reverse 12.Exit
enter choice of input: 2
enter element to delete 44

1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way.
10.Sum 11.Make the linked list reverse 12.Exit
enter choice of input: 5
87->54->NULL
number of nodes 2

1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way.
10.Sum 11.Make the linked list reverse 12.Exit
enter choice of input: 1
enter element to list: 88

1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way.
10.Sum 11.Make the linked list reverse 12.Exit
enter choice of input: 3
enter element to search 88
element found

1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way.
10.Sum 11.Make the linked list reverse 12.Exit
enter choice of input: 10
```



```
Sum of elements is 229

1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way.
10.Sum 11.Make the linked list reverse 12.Exit
enter choice of input: 11

1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way.
10.Sum 11.Make the linked list reverse 12.Exit
enter choice of input: 5
54->87->88->NULL
number of nodes 3

1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way.
10.Sum 11.Make the linked list reverse 12.Exit
enter choice of input: 7

1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way.
10.Sum 11.Make the linked list reverse 12.Exit
enter choice of input: 5
87->88->NULL
number of nodes 2

1. Insert new item. 2. Delete item. 3. Search item. 4. Insert Last. 5. Print.
6.Delete 7.Delete First 8.Insert After 9.Print Linked List in Reverse way.
10.Sum 11.Make the linked list reverse 12.Exit
enter choice of input: 12
```