

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

EXPT NO: 1A

DATE:

TITLE: Single Linked List

AIM:

Write a program of single linked list with the following operations:

- a) Create a Linked List
- b) Display the Linked List
- c) Count the number of nodes in the Linked List
- d) Search for an element and display its position
- e) Insert a node at the beginning, end, before a node, after a node and at a particular position
- f) delete a node
- g) Reverse the linked list

THEORY:

Linked list is a sequence of nodes where each node is having two parts in it. The first part is the information part while the second part of a node is a link part which is used to point to the successor node of the list if it exists, otherwise their link part should point to no node indicating that it is the terminal node of the list.

While the elements of an array occupy contiguous memory locations, those of a linked list are not constrained to be stored in adjacent locations. The order of the elements is maintained by explicit links between them.

TYPICAL NODE

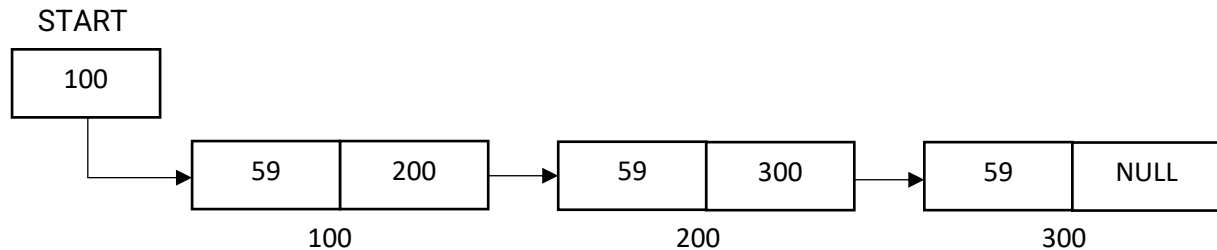
DATA	LINK
------	------

A linked list is a collection of elements called Nodes. Each Node has two sections to it. The data part and the link part. The Null in the last Node indicates that it is the last Node in the list.

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

TYPICAL LINKED LIST



SYNTAX

A Node in C may be implemented by using a self-referential structure that has a member that points to a structure of the same type.

NOTE: stdlib.h library should be included.

```
Struct node{  
    Int data;  
    Struct node *link;  
};
```

To Create A Node

```
Struct node *temp;  
temp=(struct node*)malloc(sizeof(struct node));
```

To delete the node

free() function is used

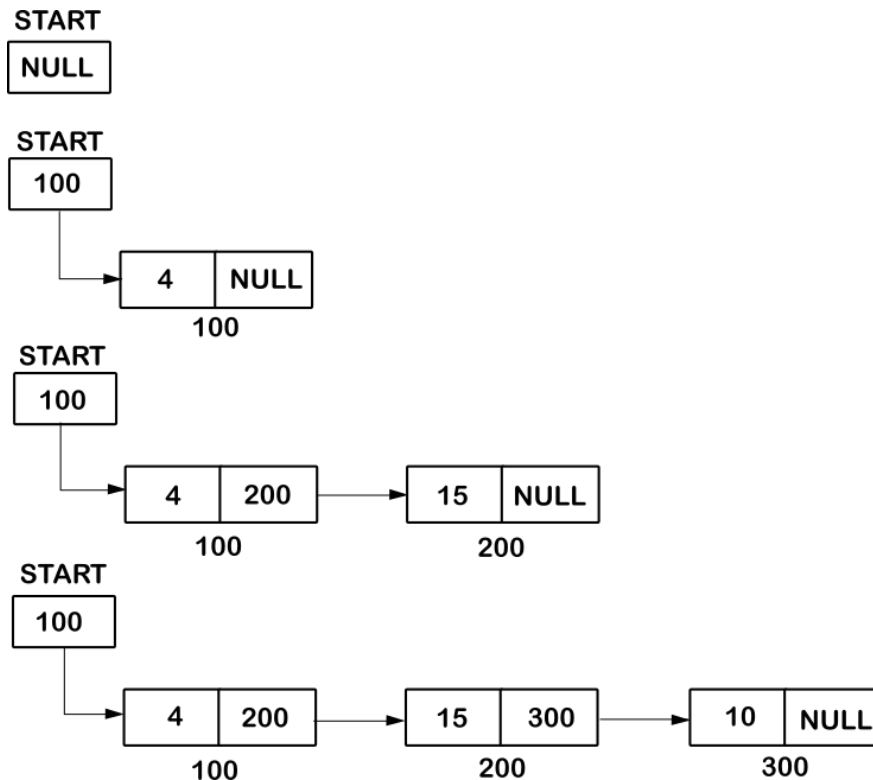
CREATING A LIST

The fundamental approach in creating a list would be to seek the number of nodes the user wants to use (n) and then utilizing this value to create that many number of nodes. It should be noted that a List can be empty or it can have a lone element either. All this node should be linked each other.

Example: Create a list which contains the following elements. 4,15,10

GOA COLLEGE OF ENGINEERING

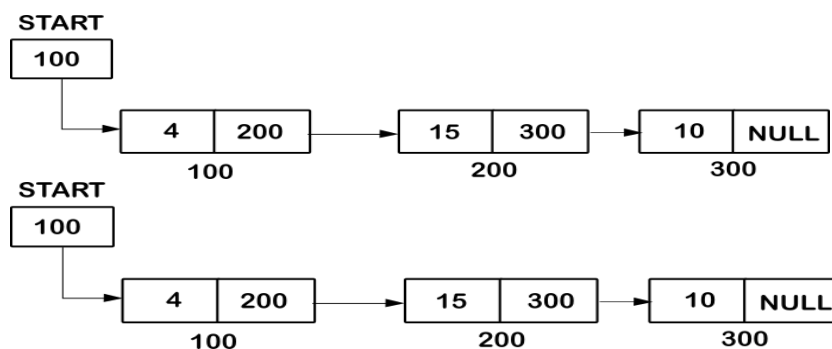
FARMAGUDI, PONDA GOA



ADD AN ELEMENT AT THE END

A new node can be appended at the end of a list by just linking the new node with the last node of the existing list, i.e the link part of the existing last node of the list will contain the address of the new node and the link section of the new node will contain NULL to indicate the end of the list.

Example: Insert the 17 at the end of the list (Previous example)



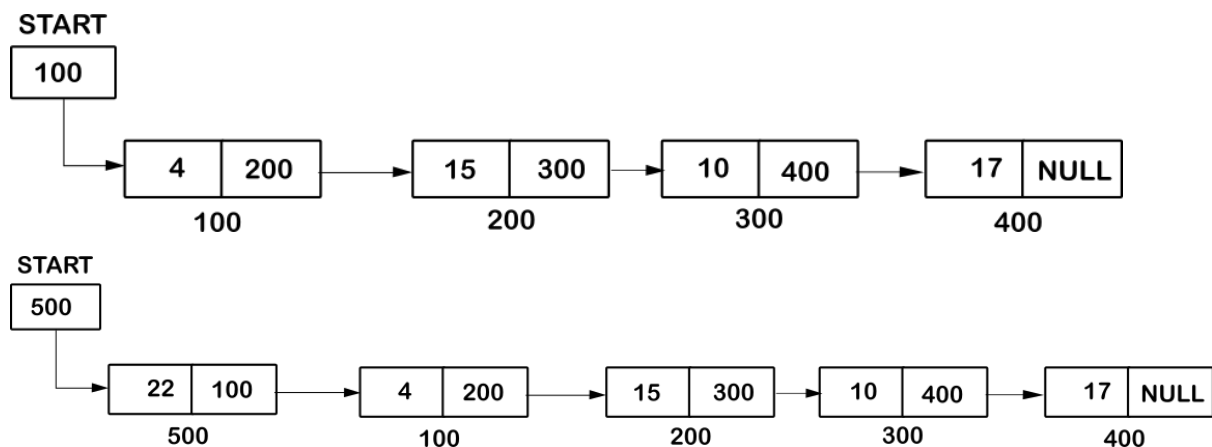
GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

ADD AN ELEMENT AT THE BEGINNING

A new node can be added at the beginning of a list by creating a new node and then storing the previous start address in the link section of the new node and then after storing the address of the new into the start pointer.

Example: Insert 22 at the beginning of the linked list (Previous Example)

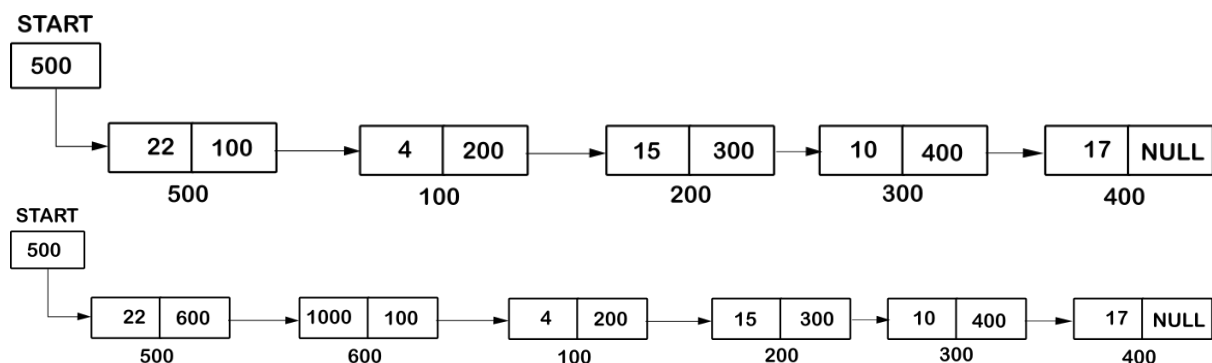


ADD AN ELEMENT BEFORE A NODE

Inserting a node before another node isn't a challenging task. It can be achieved by just breaking the existing link and joining it to the new node.

Initially a new node is created. The link section of the new node carries the address of the next node (Refer Example: Node which carries 4) and the node before it will carry the address of the new node in its link section.

Example: Insert 1000 before 4 (Previous Example)



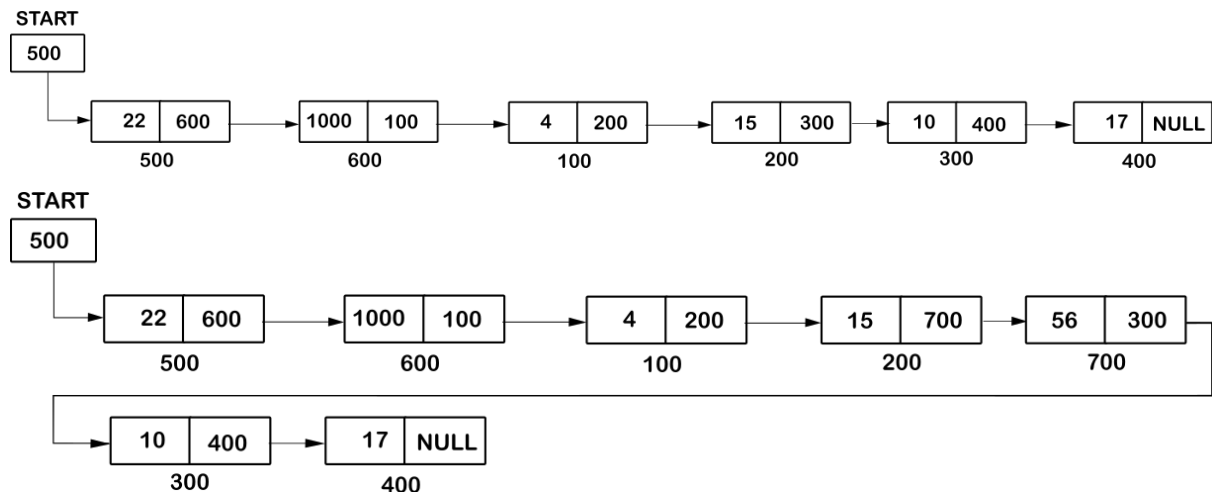
GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

ADD AN ELEMENT AFTER A NODE

To add a node after an element, a new node is created and its link section holds the address of the node which is ahead and the link section of the node before (Refer Example: Node which carries 15) it will carry the address of the new node.

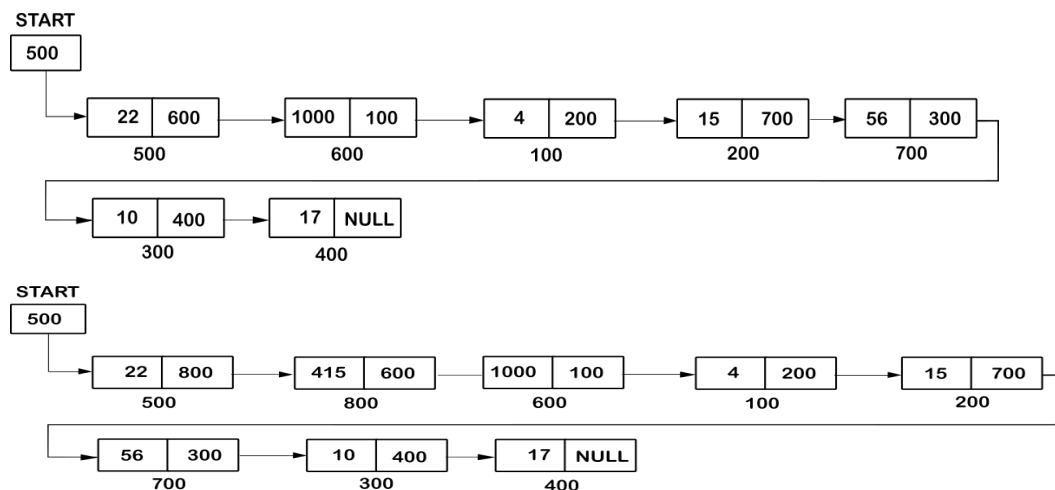
Example: Insert 56 after 15 (Previous example)



ADD AN ELEMENT AT A GIVEN POSITION

A new node is inserted at a position by creating a new node. The link section of the node contains the address of the node which previously existed at that position and the link section of the node which stands before it holds the address of the new node.

Example: Insert 415 at position 2. (Previous example)



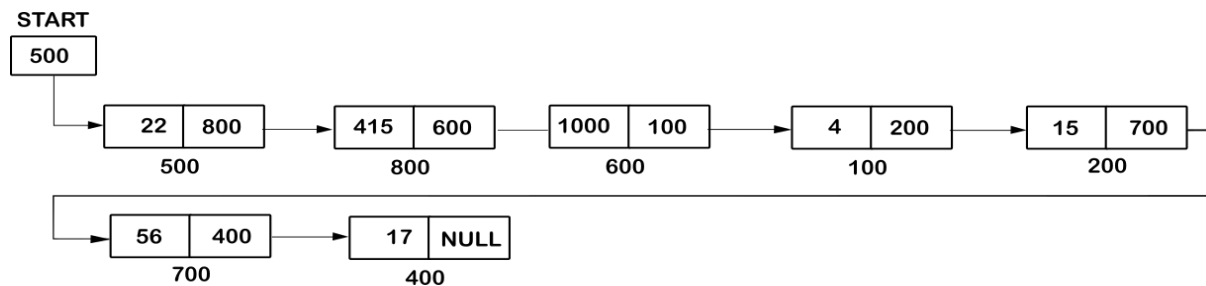
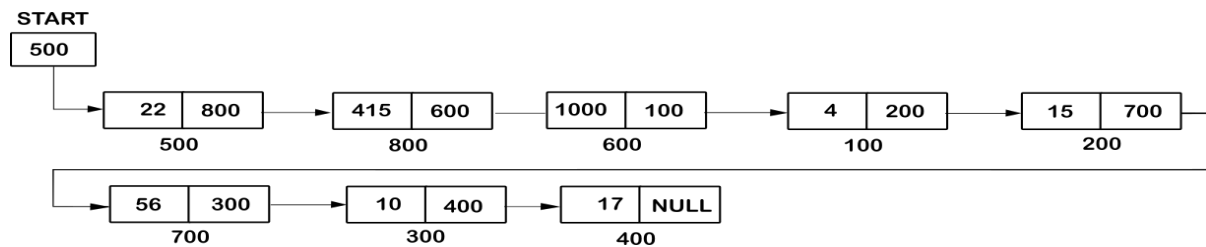
GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

DELETE AN ELEMENT

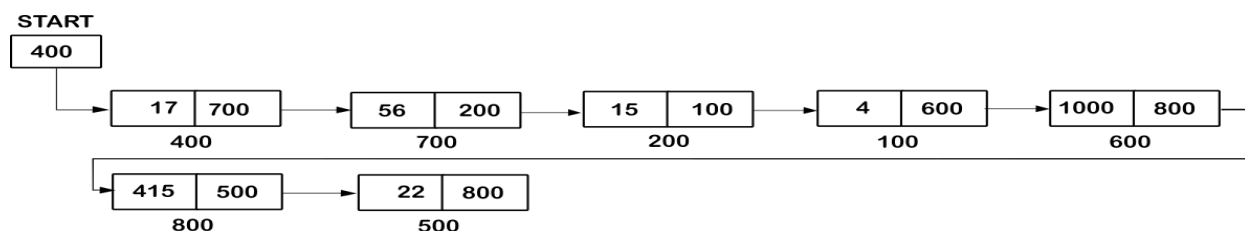
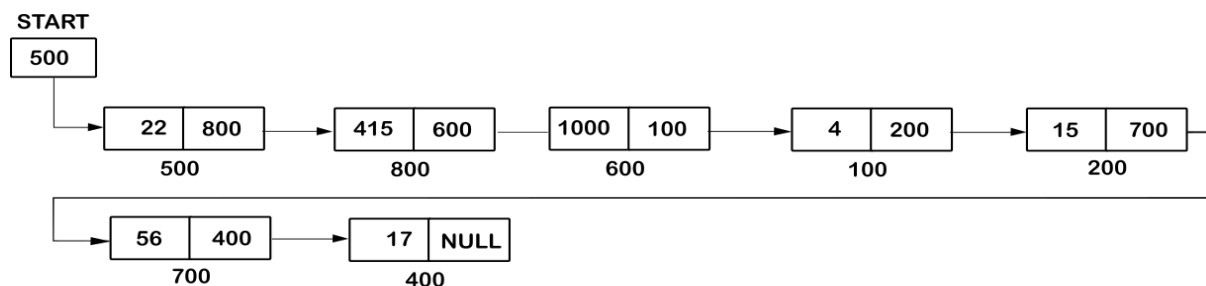
The element to be deleted is first searched throughout the list. The node prior to the node to be deleted will now carry the address which is present in the link section of the node to be deleted.

Example: Delete 10 from the linked list



REVERSING A LINKED LIST

The linked list can be reversed by just rearranging the links in the opposite direction



GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

DISPLAYING, SEARCHING AND COUNTING

All of this operation share a common aspect of traversing into the list.

The Displaying process continues till it encounters a NULL Value In the linked list.

Similarly the entire Linked list is traversed in case of counting the number of elements in the linked list. A counter is employed to keep a track of the number of elements in the List.

While in the searching process, the entire Linked list is traversed till the element is found. Once the element is found the position is noted down.

ALGORITHMS

createList()

1. Input N
2. Initialize start = NULL
3. If N=0
 1. Return start
4. Input x
5. Call addBEG(start, x)
6. for i=2....N
 1. Input x
 2. Call addEND(start, x)
7. return start

display(struct node *start)

1. Declare *p
2. if start==NULL
 1. print "List Empty"
 2. return
3. p=start
4. while p!=NULL
 1. print p->info
 2. p=p->link

addBEG(struct node *start, int x)

1. Declare *temp
2. temp->info=x
3. temp->link=start
4. start=temp
5. return start

addEND(struct node *start, int x)

1. Declare *temp,*p
2. temp->info=x
3. p=start
4. while p->link!=NULL
 1. p=p->link
5. p->link=temp
6. temp->link=NULL

count(struct node *start)

1. Declare and initialize cnt=0
2. declare *p
3. Initialize p=start
4. while p!=NULL
 1. cnt=cnt+1
 2. p=p->link
5. return start

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

Search(struct node *start)

1. Initialize pos=0
2. input v
3. declare *p
4. initialize p=start
5. while p!=NULL
 1. pos=pos+1
 2. if p->info==v
 1. print pos
 2. return
 3. p=p->link
4. print "element not found"

addPos(struct node *start)

1. Declare *temp,*p
2. declare l,pos,x
3. Input pos
4. Input x
5. temp->info=x
6. p=start
7. if pos==1
 1. temp->link=start
 2. start=temp
 3. return start
8. for(i=1;i<pos-1&&p!=NULL;i++)
 1. p=p->link
 2. if p=NULL
 1. print less elements
 - Else
 1. temp->link=p->link
 2. p->link=temp
9. return start

addBefore(struct node *start)

1. Declare *p,*temp
2. declare x and v
3. if start==NULL
 1. print "list empty"
 2. return start
4. Input x (the element before which the element Before which you want enter the element)
5. Input v (the element)
6. if start->info == x
 1. temp->info=v
 2. temp->link=start
 3. start=temp
7. p=start
8. while p->link!=NULL
 1. if p->link->info==x
 1. temp->info=v
 2. temp->link=p->link
 3. p->link=temp
 4. return start
 2. p=p->link

addAfter(struct node *start)

1. declare temp,x (int type)
2. input temp(enter the element after which you want to insert the element)
3. declare *p,*tep
4. initialize p=start
5. while p!=NULL
 1. if p->info ==temp
 1. input x (New element)
 2. tep->info=x
 3. tep->link=p->link
 4. p->link=tep
 2. p=p->link

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

del(struct node *start)

```
1. Declare *temp,*p
2.if start==NULL
    1. print "list empty"
    2. return start
3. input f (the element to be deleted)
4. if start->info==f
    1. temp=start
    2. start=start->link
    3. free(temp)
    4. Return start
5. p=start
6. while p->link!=NULL
    1. if p->Info==f
        1.temp=p->link
        2. p->link=temp->link
        3. free(temp)
        4. return start
    2. p=p->link
7. print "element not found"
8. return start
```

reverse(struct node *start)

```
1. Declare *prev,*ptr,*nxt
2. prev=NULL
3. prt=start
4. while ptr!=NULL
    1.nxt=ptr->link
    2. ptr->link=prev
    3. prev=ptr
    4. ptr=nxt
5. start=prev
6. return start
```

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

SOURCE CODE:

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

struct node{
int info;
struct node *link;
};
struct node *addEND(struct node *start,
int x)
{
struct node *temp,*p;
temp=(struct
node*)malloc(sizeof(struct node));
temp->info=x;
p=start;

while(p->link!=NULL)
p=p->link;

p->link=temp;
temp->link=NULL;

return start;
}

struct node *addBEG(struct node *start,
int x)
{
struct node *temp;
temp=(struct
node*)malloc(sizeof(struct node));

temp->info=x;
temp->link=start;
start=temp;
return start;
}

struct node *createList(struct node
*start)
{
int i,n,x;
printf("ENTER NUMBER OF
ELEMENTS TO BE ADDED\n");
scanf("%d",&n);

start=NULL;
if(n==0)
return start;

printf("ENTER THE ELEMENT: ");
scanf("%d",&x);

start=addBEG(start,x);

for(i=2;i<=n;i++)
{
printf("ENTER THE
ELEMENT: ");
scanf("%d",&x);
start=addEND(start,x);
}

return start;
}

void display(struct node *start)
{
struct node *p;
```

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

```
if(start==NULL)
{
    printf("LIST EMPTY\n");
    printf("%d ",p->info);
    return;
}
p=start;
while(p!=NULL)
{
    printf("%d ",p->info);
    p=p->link;
}
printf("\n");
}

void countList(struct node *start)
{
    struct node *p;
    int cnt=0;
    p=start;

    while(p!=NULL)
    {
        p=p->link;
        cnt++;
    }
    printf("NUMBER OF ELEMENTS:
%d\n",cnt);

    FILE *fp;
    fp=fopen("Text1.txt","a");
    fprintf(fp,"NUMBER OF
ELEMENTS: %d\n",cnt);
    fclose(fp);
}

void search(struct node *start)
{
    int v,pos=0;;

    printf("ENTER THE ELEMENT TO
BE SEARCHED\n");
    scanf("%d",&v);
    FILE *fp;
    fp=fopen("Text1.txt","a");
    struct node *p=start;
    while(p!=NULL)
    {
        pos++;
        if(p->info==v)
        {
            fprintf(fp,"ELEMENT FOUND AT
LOCATION: %d\n",pos);
            printf("ELEMENT
FOUND AT LOCATION: %d",pos);
            fclose(fp);
            return;
        }

        p=p->link;
    }
    printf("ELEMENT NOT
FOUND!\n");
    fprintf(fp,"ELEMENT NOT
FOUND!\n");
    fclose(fp);
}

void addAfter(struct node *start)
{
    int temp,x;
    printf("ENTER THE ELEMENT
AFTER WHICH YOU WANT TO INSERT
THE NEW ELEMENT: ");
    scanf("%d",&temp);
    struct node *tep,*p;
    p=start;
    while(p!=NULL)
    {
        if(p->info==temp)
        {
```

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

```
printf("ENTER THE
NEW ELEMENT: ");
scanf("%d",&x);
tep=(struct
node*)malloc(sizeof(struct node));
tep->info=x;
tep->link=p->link;
p->link=tep;
}
p=p->link;
}
}

struct node *addBefore(struct node
*start)
{
    struct node *p,*temp;
    int x,v;
    if(start==NULL)
    {
        printf("LIST EMPTY\n");
        return start;
    }
    printf("ENTER THE ELEMENT
BEFORE WHICH YOU WANT TO INSERT
THE NEW ELEMENT: ");
    scanf("%d",&x);
    printf("ENTER THE ELEMENT
YOU WANT TO ENTER\n");
    scanf("%d",&v);
    temp=(struct
node*)malloc(sizeof(struct node));
    if(start->info==x)
    {
        ;
        temp->info=v;
        temp->link=start;
        start=temp;
    }
    p=start;
    while(p->link!=NULL)
    {
        if(p->link->info==x)
        {
            temp->info=v;
            temp->link=p->link;
            p->link=temp;
        }
    }
}

struct node *addPos(struct node *start)
{
    struct node *temp,*p;
    int i,pos,x;
    temp=(struct
node*)malloc(sizeof(struct node));
    printf("ENTER THE POSITIONS:
");
    scanf("%d",&pos);
    printf("ENTER THE
ELEMENT\n");
    scanf("%d",&x);
    temp->info=x;

    if(pos==1)
    {
        temp->link=start;
        start=temp;
        return start;
    }
    p=start;
    for(i=1;i<pos-1&&p!=NULL;i++)
    {
        p=p->link;
        if(p==NULL)
            printf("THERE ARE LESS
THAN %d ELEMENTS\n",pos);
    }
    temp->link=p->link;
    p->link=temp;
}
```

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

```
        return start;
    }

    struct node *del(struct node *start)
    {
        struct node *temp,*p;
        int f;
        if(start==NULL)
        {
            printf("LIST EMPTY\n");
            return start;
        }

        printf("ENTER THE ELEMENT TO
        BE DELETED\n");
        scanf("%d",&f);

        if(start->info==f)
        {
            temp=start;
            start=start->link;
            free(temp);
            return start;
        }
        p=start;
        while(p->link!=NULL)
        {
            if(p->link->info==f)
            {
                temp=p->link;
                p->link=temp->link;
                free(temp);
                return start;
            }
            p=p->link;
        }
    }

    int main()
    {
        int choice,y;
        struct node *start,*p;
        FILE *fp;

        printf("ELEMENT NOT
        FOUND\n");
        return start;
    }

    struct node *reverse(struct node *start)
    {
        struct node *prev, *ptr, *nxt;
        prev=NULL;
        ptr=start;

        while(ptr!=NULL)
        {
            nxt=ptr->link;
            ptr->link=prev;
            prev=ptr;
            ptr=nxt;
        }
        start=prev;
        return start;
    }

    void wri(struct node *start)
    {
        FILE *fp;
        struct node *p;
        fp=fopen("Text1.txt","a");
        p=start;
        while(p!=NULL)
        {
            fprintf(fp,"%d ",p-
            >info);

            p=p->link;
        }
        fprintf(fp,"\n");
        fclose(fp);
    }
}
```

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

```
do{
    printf(" \n\n***** LINKED LIST *****\n");
    printf(" ENTER THE CORRESPONDING NUMBER TO ACCESS IT\n\n");
    printf("-> CREATE A LIST.....1\n");
    printf("-> DISPLAY THE LIST.....2\n");
    printf("-> COUNT THE NUMBER OF ELEMENTS.....3\n");
    printf("-> SEARCH AN ELEMENT IN THE LIST.....4\n");
    printf("-> ADD AN ELEMENT AT THE BEGINNING.....5\n");
    printf("-> ADD AN ELEMENT AT THE END.....6\n");
    printf("-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7\n");
    printf("-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8\n");
    printf("-> ADD AN ELEMENT AT A POSITION.....9\n");
    printf("-> DELETE AN ELEMENT.....10\n");
    printf("-> REVERSE THE LIST.....11\n");
    printf("-> EXIT.....0\n");

    printf("ENTER YOUR CHOICE: ");
    scanf("%d",&choice);
    printf("\n\n");
    switch(choice)
    {
        case 1:
            start=createList(start);

            fp=fopen("Text1.txt","w");
            p=start;
            while(p!=NULL)
            {
                fprintf(fp,"%d ",p->info);
                p=p->link;
            }
            fprintf(fp,"\n");
            fclose(fp);
            printf("ELEMENTS ADDED SUCCESSFULLY\n");

            break;

        case 2:
            display(start);
            break;
```

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

case 3:

```
countList(start);
```

```
break;
```

case 4:

```
search(start);
```

```
break;
```

case 5:

```
printf("ENTER THE ELEMENT: ");
```

```
scanf("%d",&y);
```

```
start=addBEG(start,y);
```

```
wri(start);
```

```
break;
```

case 6:

```
printf("ENTER THE ELEMENT: ");
```

```
scanf("%d",&y);
```

```
start=addEND(start,y);
```

```
wri(start);
```

```
break;
```

case 7:

```
start=addBefore(start);
```

```
wri(start);
```

```
break;
```

case 8:

```
addAfter(start);
```

```
wri(start);
```

```
break;
```

case 9:

```
start=addPos(start);
```

```
wri(start);
```

```
break;
```

case 10:

```
start=del(start);
```

```
wri(start);
```

```
break;
```

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

```
        case 11:
            start=reverse(start);
            wri(start);
            break;

        case 0:
            return 0;
        default:
            printf("INVALIDOPTION\n");
    }

    }while(1);

}
```


GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

OUTPUT

```
"C:\Users\Lloyd\Desktop\DS\expt4.exe"

***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 1

ENTER NUMBER OF ELEMENTS TO BE ADDED
4
ENTER THE ELEMENT: 4
ENTER THE ELEMENT: 10
ENTER THE ELEMENT: 15
ENTER THE ELEMENT: 17
ELEMENTS ADDED SUCCESSFULLY
```

```
***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 2
```

4 10 15 17

```
***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 3
```

NUMBER OF ELEMENTS: 4

```
***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 4
```

```
ENTER THE ELEMENT TO BE SEARCHED
15
ELEMENT FOUND AT LOCATION: 3
```

```
***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 5
```

ENTER THE ELEMENT: 200

```
***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 6
```

ENTER THE ELEMENT: 300

```
***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 7
```

ENTER THE ELEMENT BEFORE WHICH YOU WANT TO INSERT THE NEW ELEMENT: 15
ENTER THE ELEMENT YOU WANT TO ENTER
450

```
***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 8
```

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA

ENTER THE ELEMENT AFTER WHICH YOU WANT TO INSERT THE NEW ELEMENT: 10
ENTER THE NEW ELEMENT: 922

***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 9

ENTER THE POSITIONS: 2
ENTER THE ELEMENT
500

***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 10

ENTER THE ELEMENT TO BE DELETED
1
ELEMENT NOT FOUND

***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 10

ENTER THE ELEMENT TO BE DELETED
1
ELEMENT NOT FOUND

***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 10

ENTER THE ELEMENT TO BE DELETED
500

***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 11

***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: 2

300 17 15 450 922 10 4 200

***** LINKED LIST *****
ENTER THE CORRESPONDING NUMBER TO ACCESS IT

-> CREATE A LIST.....1
-> DISPLAY THE LIST.....2
-> COUNT THE NUMBER OF ELEMENTS.....3
-> SEARCH AN ELEMENT IN THE LIST.....4
-> ADD AN ELEMENT AT THE BEGINNING.....5
-> ADD AN ELEMENT AT THE END.....6
-> ADD AN ELEMENT BEFORE ANOTHER ELEMENT.....7
-> ADD AN ELEMENT AFTER ANOTHER ELEMENT.....8
-> ADD AN ELEMENT AT A POSITION.....9
-> DELETE AN ELEMENT.....10
-> REVERSE THE LIST.....11
-> EXIT.....0
ENTER YOUR CHOICE: █

GOA COLLEGE OF ENGINEERING

FARMAGUDI, PONDA GOA



Text1 - Notepad

File Edit Format View Help

4 10 15 17

NUMBER OF ELEMENTS: 4

ELEMENT FOUND AT LOCATION: 3

200 4 10 15 17

200 4 10 15 17 300

200 4 10 450 15 17 300

200 4 10 922 450 15 17 300

200 500 4 10 922 450 15 17 300

200 500 4 10 922 450 15 17 300

200 4 10 922 450 15 17 300

300 17 15 450 922 10 4 200

CONCLUSION:

The Given problem statement was successfully compiled and executed.

Findings:

Linked Lists are dynamic. Their size can be increased or decreased during run time and hence can be an optimum choice over arrays. Also, the space is well utilized in case of Linked List.

Compilation time: 0.20 Secs

Learnings

- 1) Concept of Linked Lists
- 2) Implementation of Linked Lists