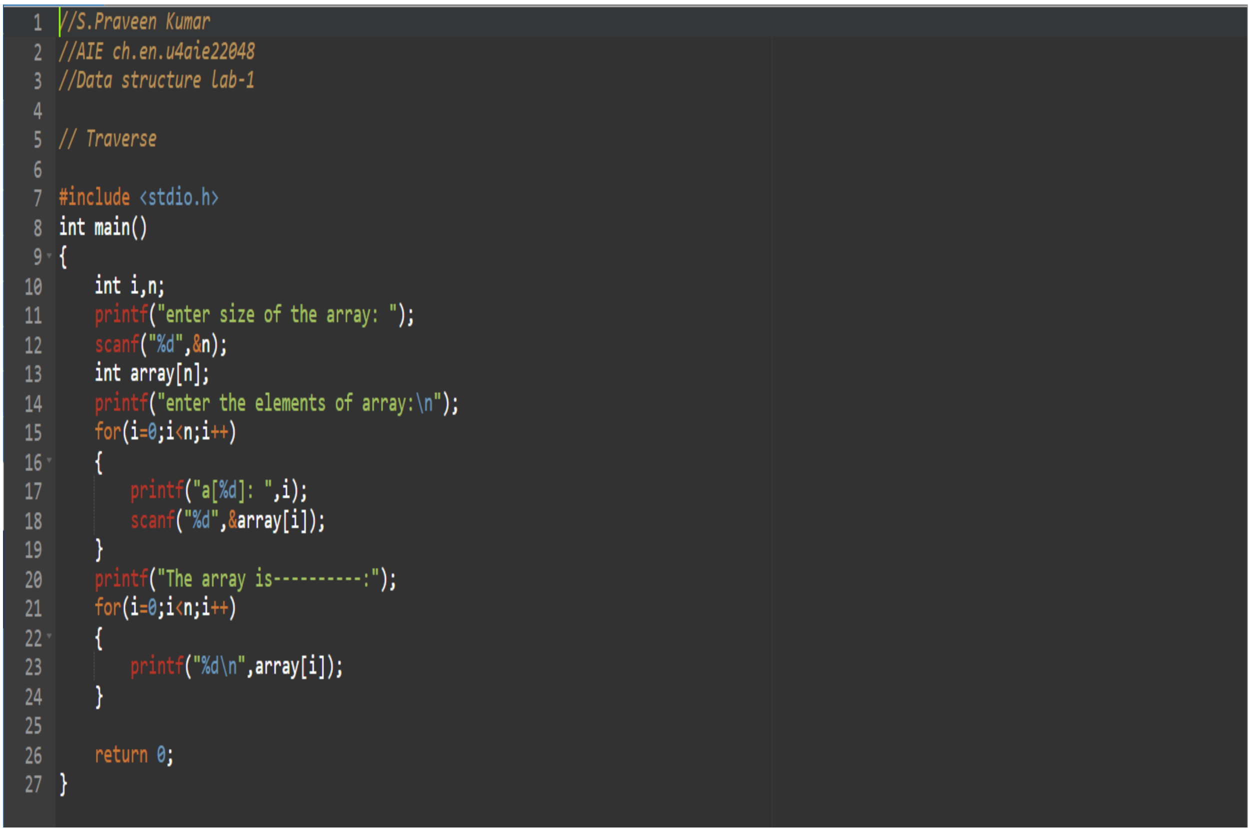
DATA STRUCTURE – 1

LAB-1

S.Praveen kumar
ch.en.u4aie22048

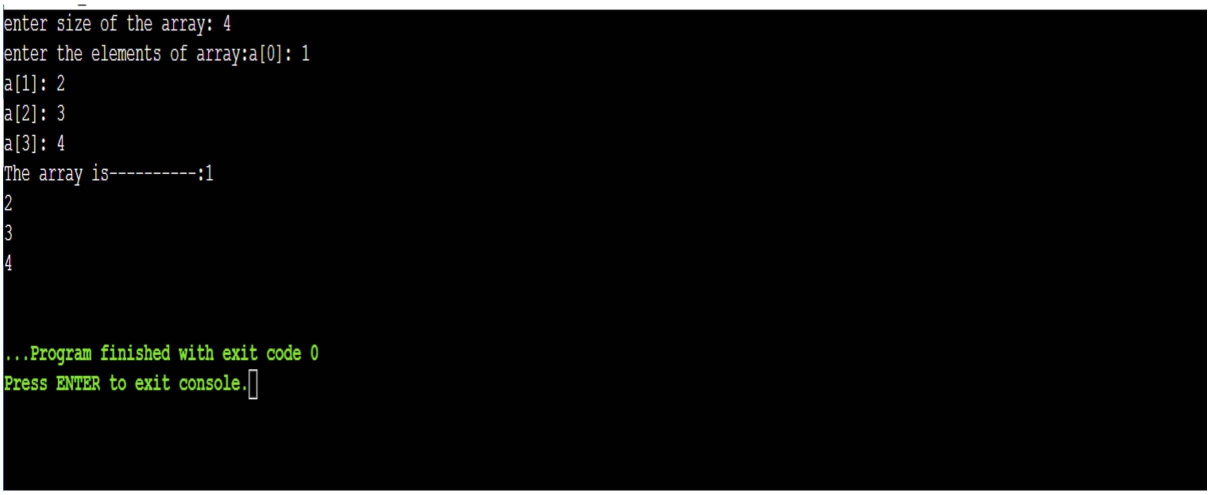
Traverse

Program:



```
//S.Praveen Kumar
//AIE ch.en.u4aie22048
//Data structure lab-1

// Traverse

#include <stdio.h>
int main()
{
    int i,n;
    printf("enter size of the array: ");
    scanf("%d",&n);
    int array[n];
    printf("enter the elements of array:\n");
    for(i=0;i<n;i++)
    {
        printf("a[%d]: ",i);
        scanf("%d",&array[i]);
    }
    printf("The array is----------:");
    for(i=0;i<n;i++)
    {
        printf("%d\n",array[i]);
    }

    return 0;
}
```

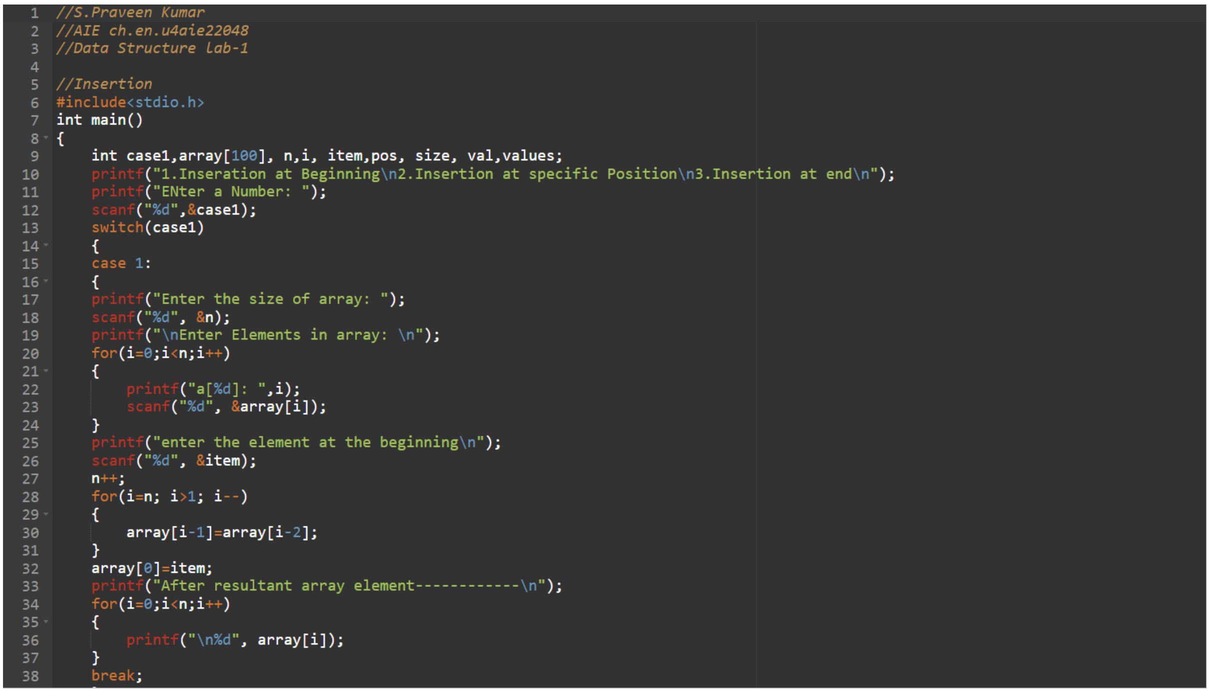
Output:



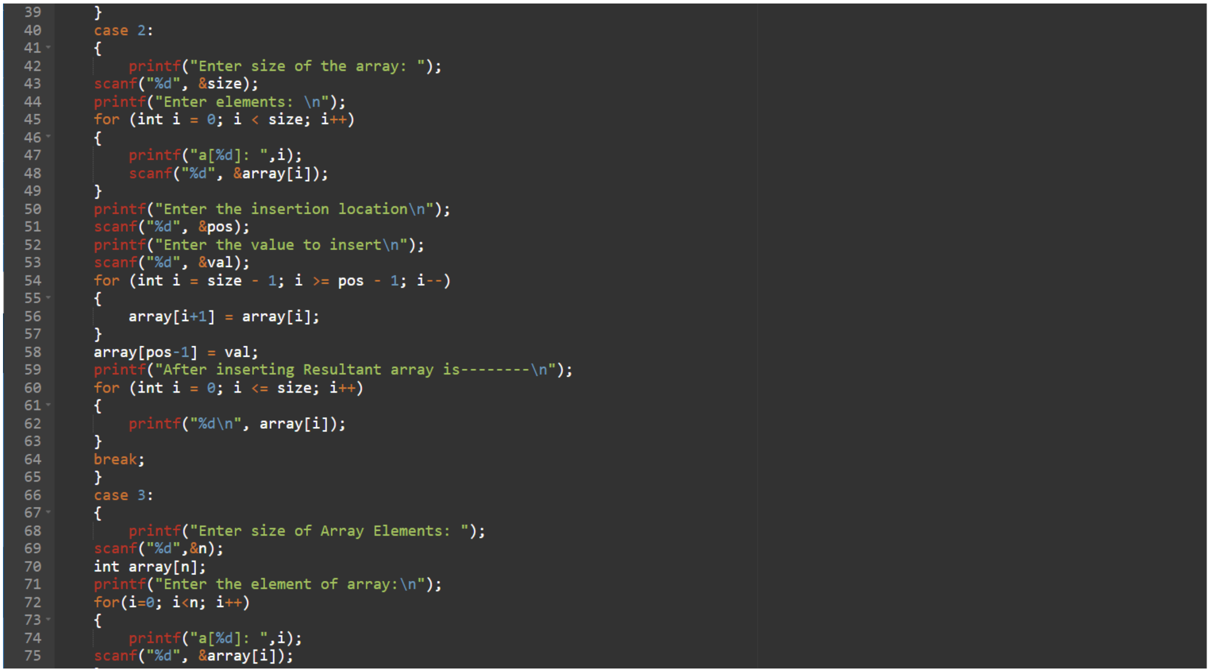
```
enter size of the array: 4
enter the elements of array:a[0]: 1
a[1]: 2
a[2]: 3
a[3]: 4
The array is----------:1
2
3
4


...Program finished with exit code 0
Press ENTER to exit console.
```

Insertion:

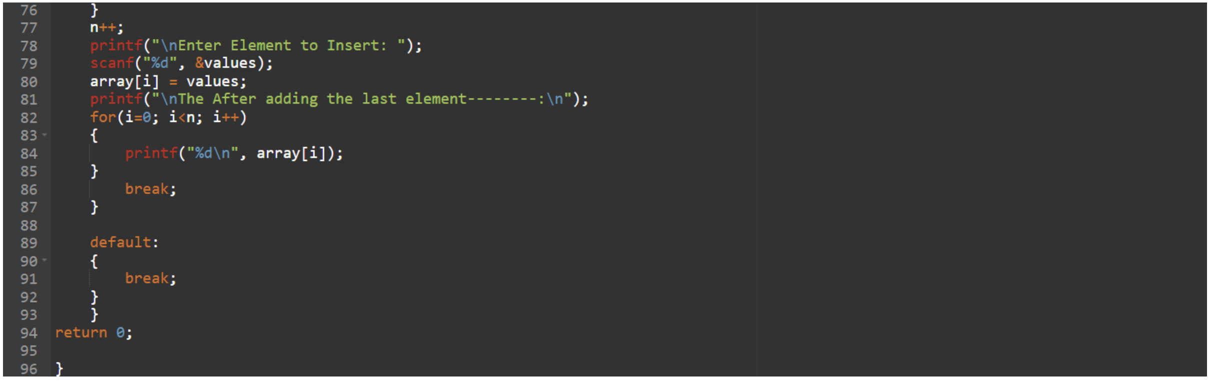
Program:



```
//S.Praveen Kumar
//AIE ch.en.u4aie22048
//Data Structure lab-1

//Insertion
#include<stdio.h>
int main()
{
    int case1,array[100], n,i, item,pos, size, val,values;
    printf("1.Inseration at Beginning\n2.Insertion at specific Position\n3.Insertion at end\n");
    printf("ENter a Number: ");
    scanf("%d",&case1);
    switch(case1)
    {
    case 1:
    {
    printf("Enter the size of array: ");
    scanf("%d", &n);
    printf("\nEnter Elements in array: \n");
    for(i=0;i<n;i++)
    {
        printf("a[%d]: ",i);
        scanf("%d", &array[i]);
    }
    printf("enter the element at the beginning\n");
    scanf("%d", &item);
    n++;
    for(i=n; i>1; i--)
    {
        array[i-1]=array[i-2];
    }
    array[0]=item;
    printf("After resultant array element-----------\n");
    for(i=0;i<n;i++)
    {
        printf("\n%d", array[i]);
    }
    break;
```

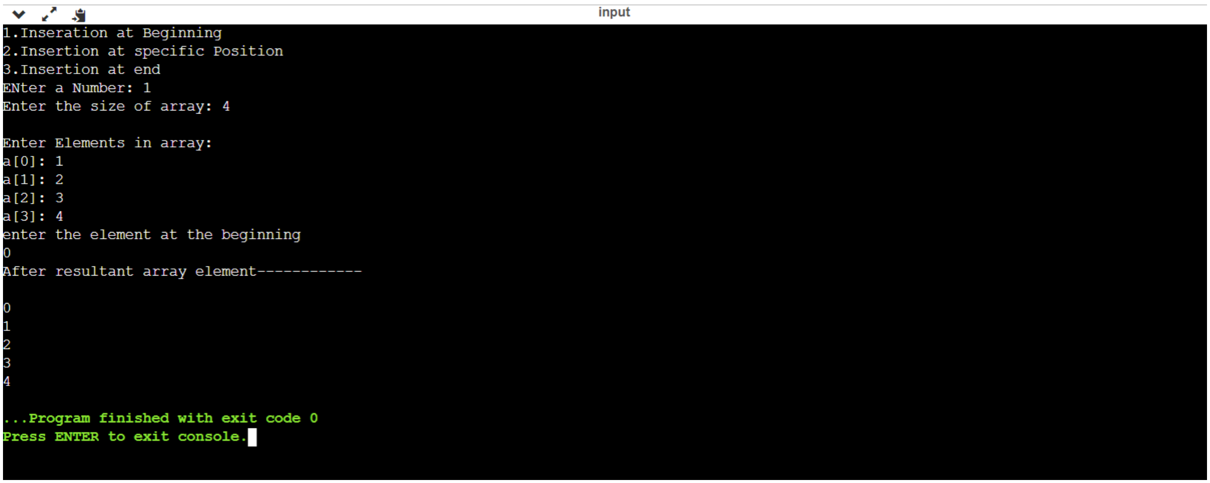


```
    }
    case 2:
    {
        printf("Enter size of the array: ");
    scanf("%d", &size);
    printf("Enter elements: \n");
    for (int i = 0; i < size; i++)
    {
        printf("a[%d]: ",i);
        scanf("%d", &array[i]);
    }
    printf("Enter the insertion location\n");
    scanf("%d", &pos);
    printf("Enter the value to insert\n");
    scanf("%d", &val);
    for (int i = size - 1; i >= pos - 1; i--)
    {
        array[i+1] = array[i];
    }
    array[pos-1] = val;
    printf("After inserting Resultant array is--------\n");
    for (int i = 0; i <= size; i++)
    {
        printf("%d\n", array[i]);
    }
    break;
    }
    case 3:
    {
        printf("Enter size of Array Elements: ");
    scanf("%d",&n);
    int array[n];
    printf("Enter the element of array:\n");
    for(i=0; i<n; i++)
    {
        printf("a[%d]: ",i);
    scanf("%d", &array[i]);
```

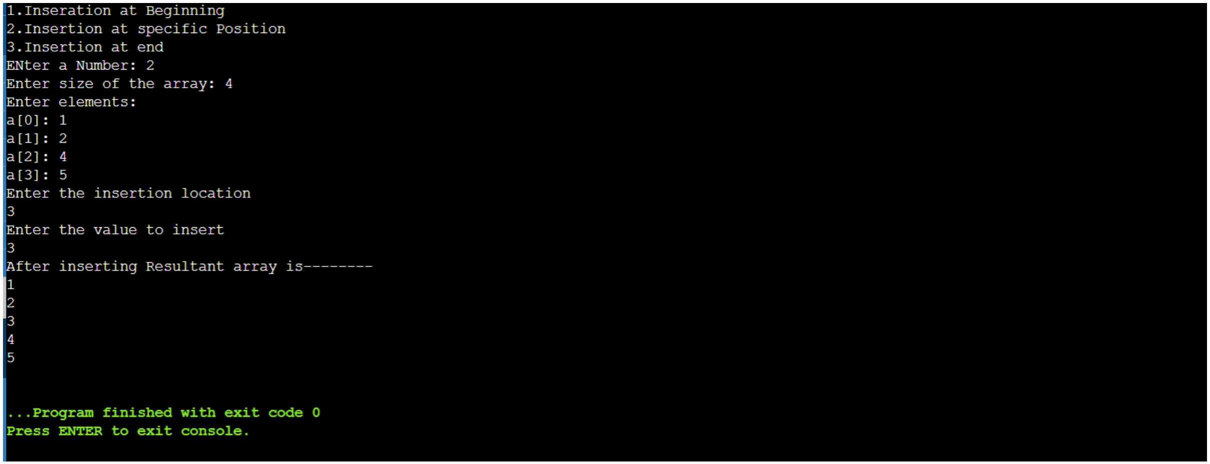


```
    }
    n++;
    printf("\nEnter Element to Insert: ");
    scanf("%d", &values);
    array[i] = values;
    printf("\nThe After adding the last element--------:\n");
    for(i=0; i<n; i++)
    {
        printf("%d\n", array[i]);
    }
        break;
    }

    default:
    {
        break;
    }
    }
return 0;

}
```

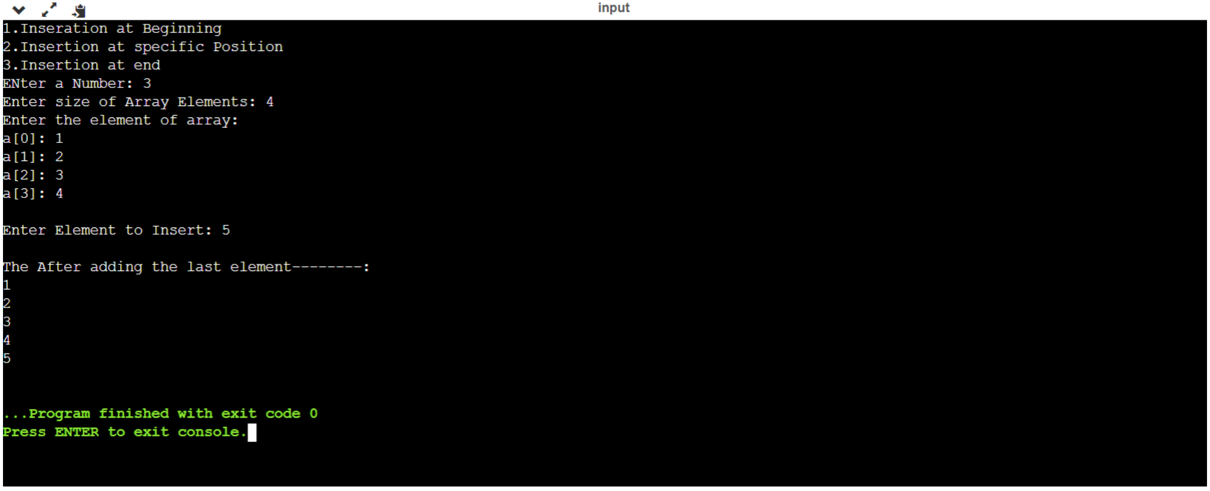
Output:



```
1.Inseration at Beginning
2.Insertion at specific Position
3.Insertion at end
ENter a Number: 1
Enter the size of array: 4

Enter Elements in array:
a[0]: 1
a[1]: 2
a[2]: 3
a[3]: 4
enter the element at the beginning
0
After resultant array element------------

0
1
2
3
4

...Program finished with exit code 0
Press ENTER to exit console.
```



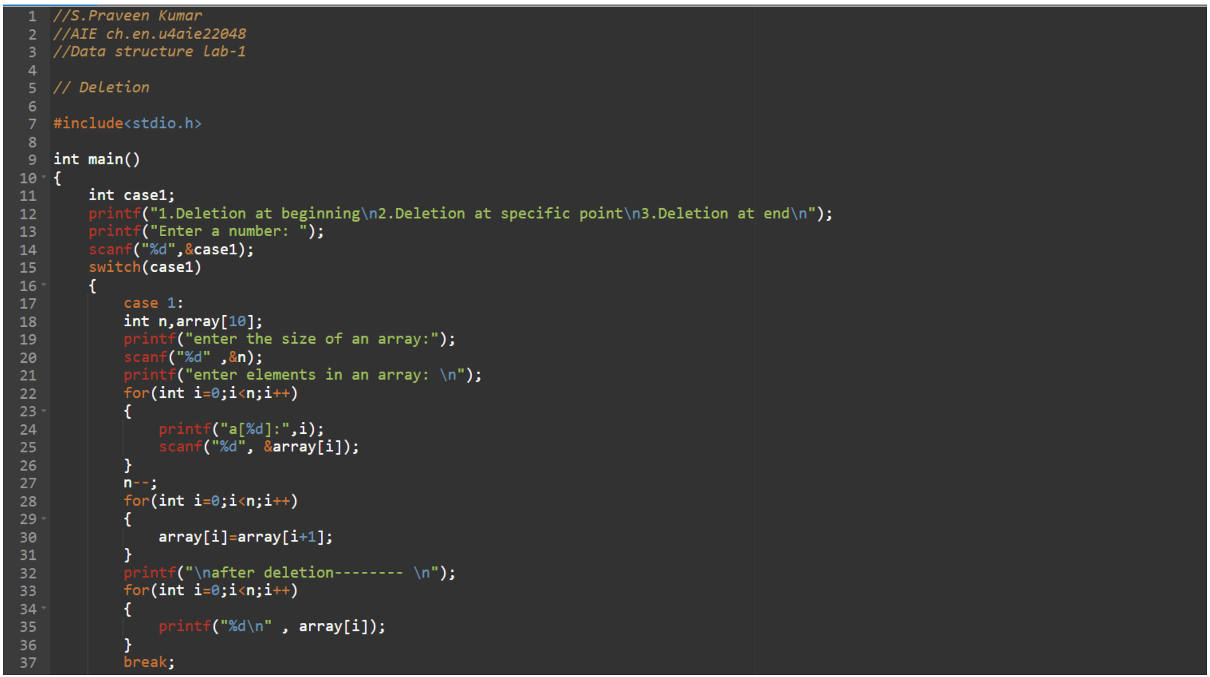
```
1.Inseration at Beginning
2.Insertion at specific Position
3.Insertion at end
ENter a Number: 2
Enter size of the array: 4
Enter elements:
a[0]: 1
a[1]: 2
a[2]: 4
a[3]: 5
Enter the insertion location
3
Enter the value to insert
3
After inserting Resultant array is--------
1
2
3
4
5


...Program finished with exit code 0
Press ENTER to exit console.
```



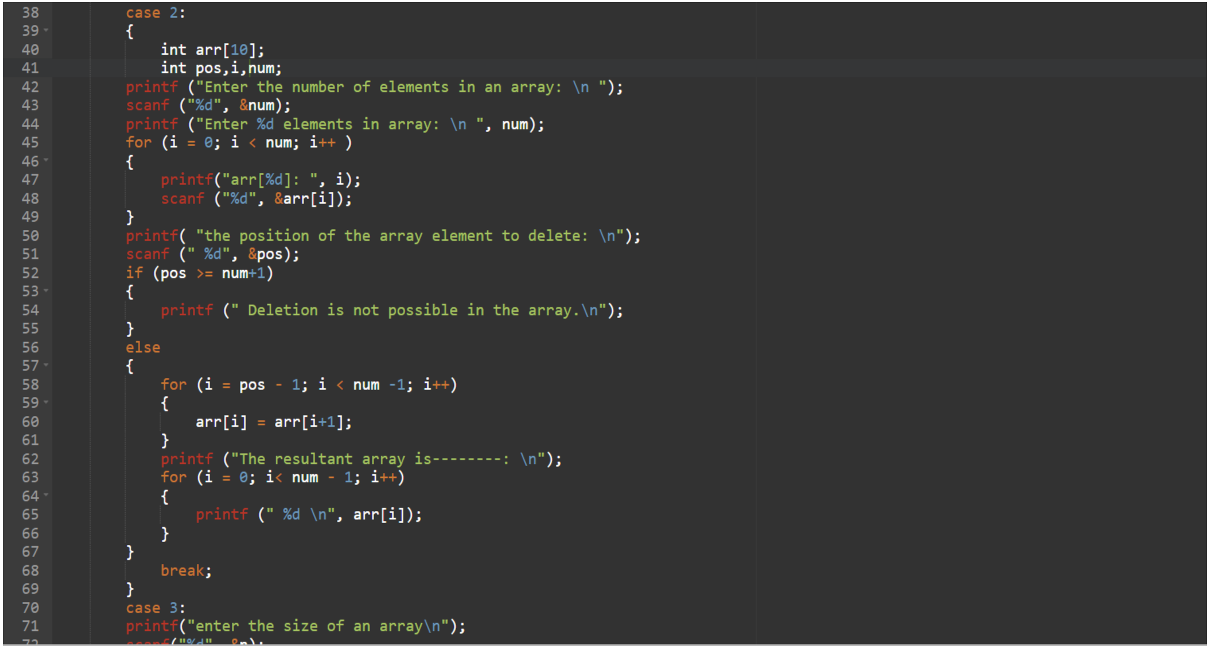
```
1.Inseration at Beginning
2.Insertion at specific Position
3.Insertion at end
ENter a Number: 3
Enter size of Array Elements: 4
Enter the element of array:
a[0]: 1
a[1]: 2
a[2]: 3
a[3]: 4

Enter Element to Insert: 5

The After adding the last element--------:
1
2
3
4
5


...Program finished with exit code 0
Press ENTER to exit console.
```

Deletion:

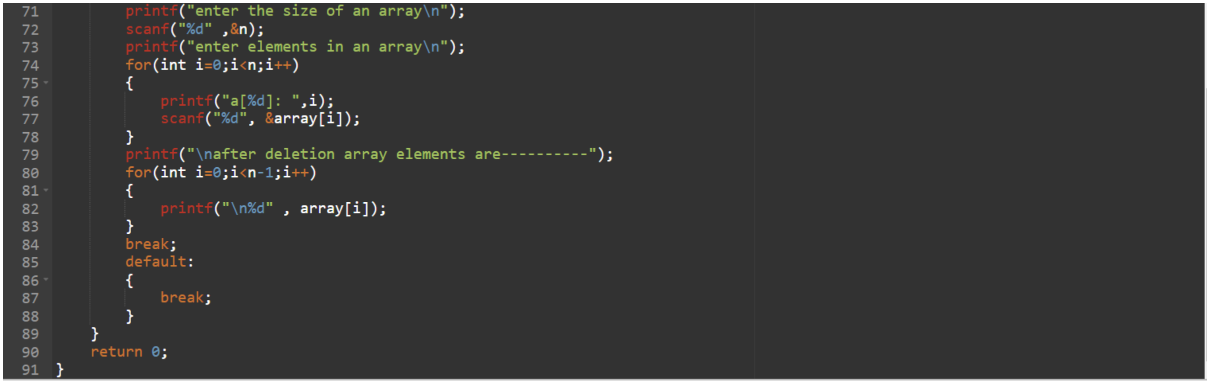
Program:



```c
//S.Praveen Kumar
//AIE ch.en.u4aie22048
//Data structure lab-1

// Deletion

#include<stdio.h>

int main()
{
    int case1;
    printf("1.Deletion at beginning\n2.Deletion at specific point\n3.Deletion at end\n");
    printf("Enter a number: ");
    scanf("%d",&case1);
    switch(case1)
    {
        case 1:
        int n,array[10];
        printf("enter the size of an array:");
        scanf("%d" ,&n);
        printf("enter elements in an array: \n");
        for(int i=0;i<n;i++)
        {
            printf("a[%d]:",i);
            scanf("%d", &array[i]);
        }
        n--;
        for(int i=0;i<n;i++)
        {
            array[i]=array[i+1];
        }
        printf("\nafter deletion-------- \n");
        for(int i=0;i<n;i++)
        {
            printf("%d\n" , array[i]);
        }
        break;
```



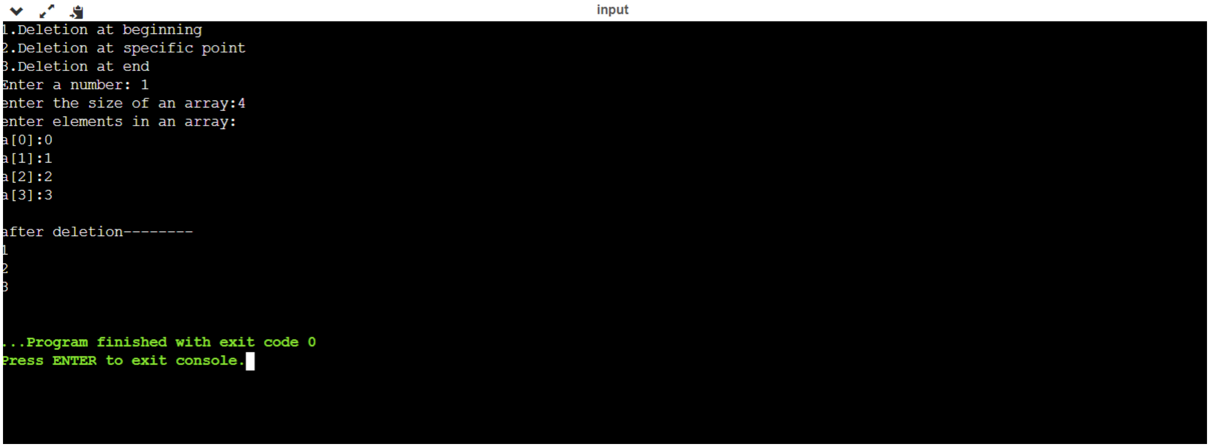
```c
        case 2:
        {
            int arr[10];
            int pos,i,num;
        printf ("Enter the number of elements in an array: \n ");
        scanf ("%d", &num);
        printf ("Enter %d elements in array: \n ", num);
        for (i = 0; i < num; i++ )
        {
            printf("arr[%d]: ", i);
            scanf ("%d", &arr[i]);
        }
        printf( "the position of the array element to delete: \n");
        scanf (" %d", &pos);
        if (pos >= num+1)
        {
            printf (" Deletion is not possible in the array.\n");
        }
        else
        {
            for (i = pos - 1; i < num -1; i++)
            {
                arr[i] = arr[i+1];
            }
            printf ("The resultant array is--------: \n");
            for (i = 0; i< num - 1; i++)
            {
                printf (" %d \n", arr[i]);
            }
        }
            break;
        }
        case 3:
        printf("enter the size of an array\n");
```



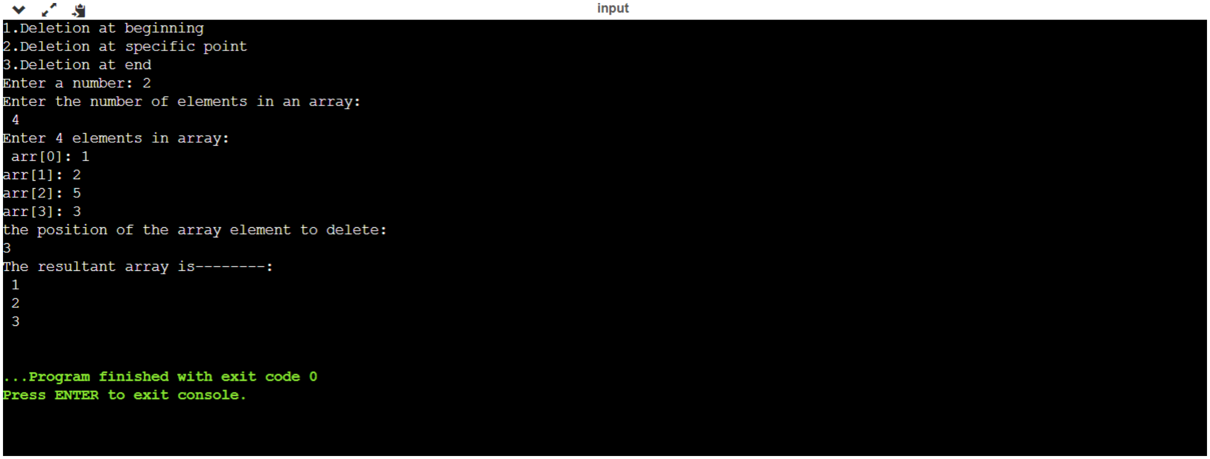
```c
        printf("enter the size of an array\n");
```

```c
        scanf("%d" ,&n);
        printf("enter elements in an array\n");
        for(int i=0;i<n;i++)
        {
            printf("a[%d]: ",i);
            scanf("%d", &array[i]);
        }
        printf("\nafter deletion array elements are----------");
        for(int i=0;i<n-1;i++)
        {
            printf("\n%d" , array[i]);
        }
        break;
        default:
        {
            break;
        }
    }
    return 0;
}
```

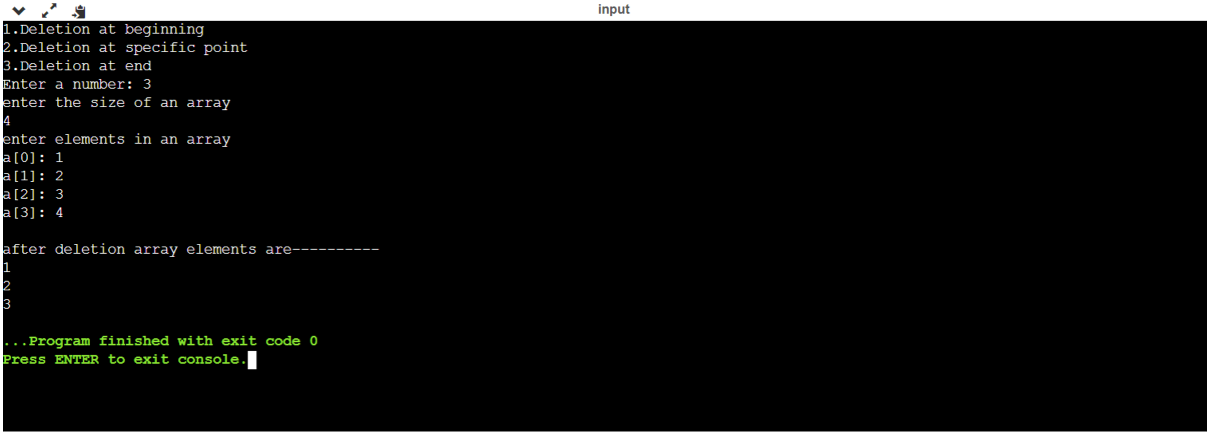
Output:



```
1.Deletion at beginning
2.Deletion at specific point
3.Deletion at end
Enter a number: 1
enter the size of an array:4
enter elements in an array:
a[0]:0
a[1]:1
a[2]:2
a[3]:3

after deletion--------
1
2
3


...Program finished with exit code 0
Press ENTER to exit console.
```



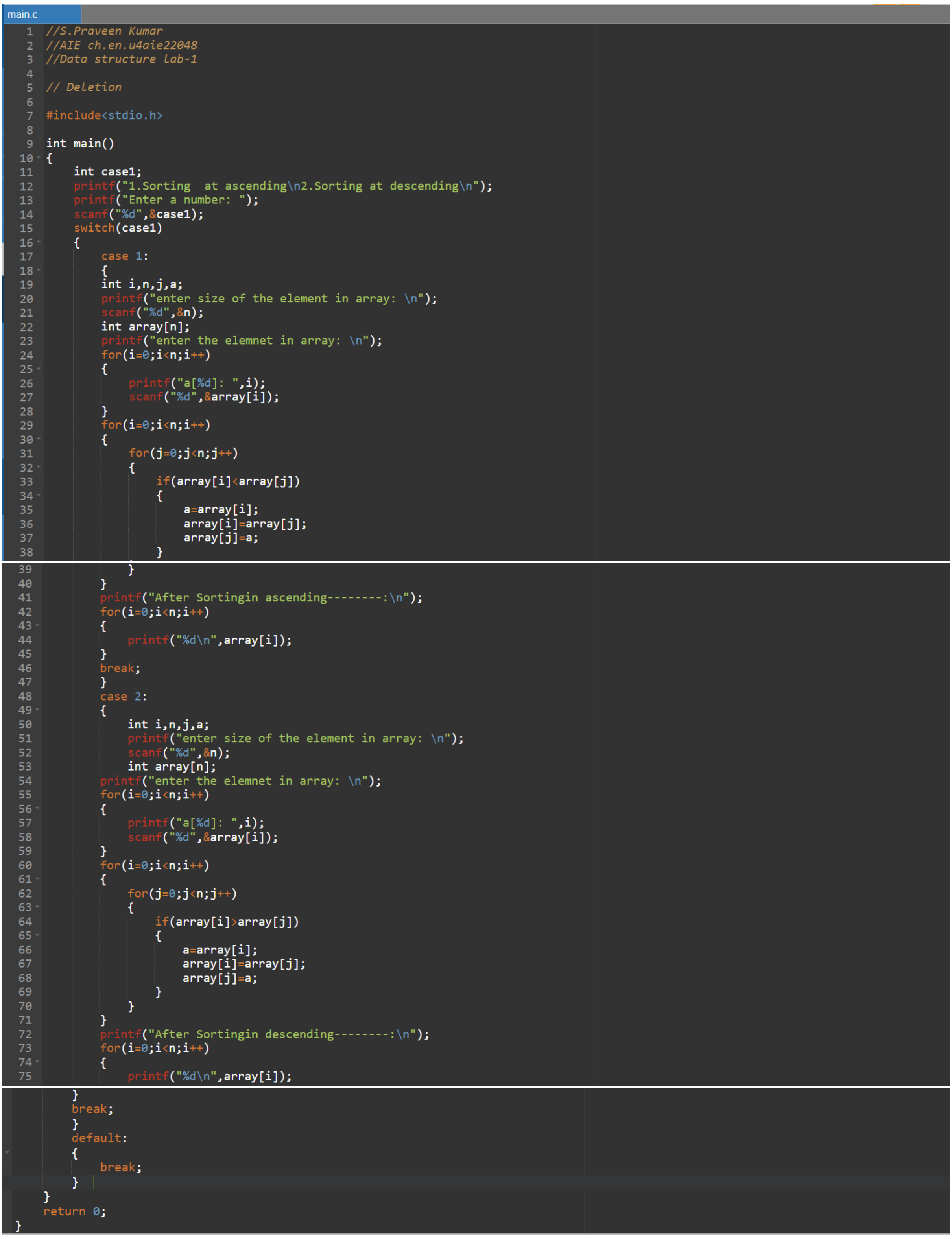
```
1.Deletion at beginning
2.Deletion at specific point
3.Deletion at end
Enter a number: 2
Enter the number of elements in an array:
 4
Enter 4 elements in array:
 arr[0]: 1
arr[1]: 2
arr[2]: 5
arr[3]: 3
the position of the array element to delete:
3
The resultant array is--------:
 1
 2
 3


...Program finished with exit code 0
Press ENTER to exit console.
```



```
1.Deletion at beginning
2.Deletion at specific point
3.Deletion at end
Enter a number: 3
enter the size of an array
4
enter elements in an array
a[0]: 1
a[1]: 2
a[2]: 3
a[3]: 4

after deletion array elements are----------
1
2
3

...Program finished with exit code 0
Press ENTER to exit console.
```

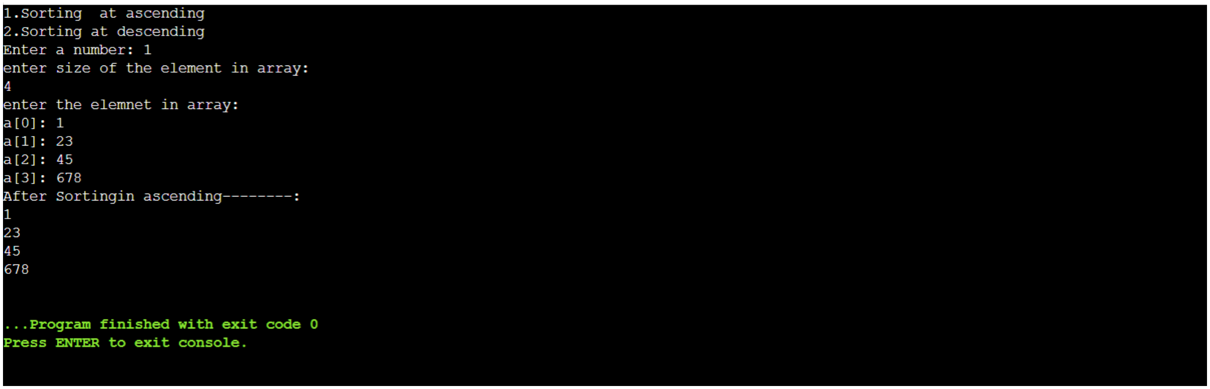
Sorting

Program:

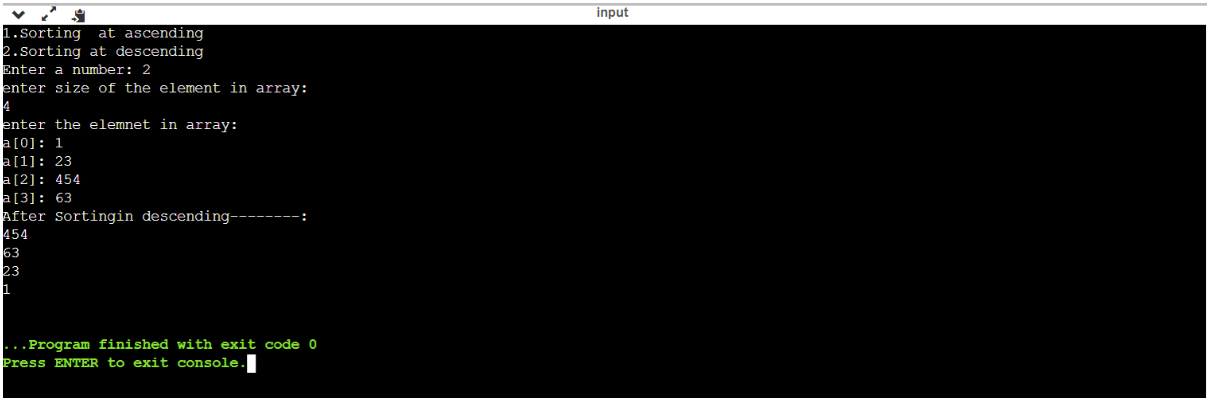


```c
//S.Praveen Kumar
//AIE ch.en.u4aie22048
//Data structure lab-1

// Deletion

#include<stdio.h>

int main()
{
    int case1;
    printf("1.Sorting  at ascending\n2.Sorting at descending\n");
    printf("Enter a number: ");
    scanf("%d",&case1);
    switch(case1)
    {
        case 1:
        {
        int i,n,j,a;
        printf("enter size of the element in array: \n");
        scanf("%d",&n);
        int array[n];
        printf("enter the elemnet in array: \n");
        for(i=0;i<n;i++)
        {
            printf("a[%d]: ",i);
            scanf("%d",&array[i]);
        }
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;j++)
            {
                if(array[i]<array[j])
                {
                    a=array[i];
                    array[i]=array[j];
                    array[j]=a;
                }
            }
        }
        printf("After Sortingin ascending--------:\n");
        for(i=0;i<n;i++)
        {
            printf("%d\n",array[i]);
        }
        break;
        }
        case 2:
        {
            int i,n,j,a;
            printf("enter size of the element in array: \n");
            scanf("%d",&n);
            int array[n];
        printf("enter the elemnet in array: \n");
        for(i=0;i<n;i++)
        {
            printf("a[%d]: ",i);
            scanf("%d",&array[i]);
        }
        for(i=0;i<n;i++)
        {
            for(j=0;j<n;j++)
            {
                if(array[i]>array[j])
                {
                    a=array[i];
                    array[i]=array[j];
                    array[j]=a;
                }
            }
        }
        printf("After Sortingin descending--------:\n");
        for(i=0;i<n;i++)
        {
            printf("%d\n",array[i]);
        }
        break;
        }
        default:
        {
            break;
        }
    }
    return 0;
}
```

Output:



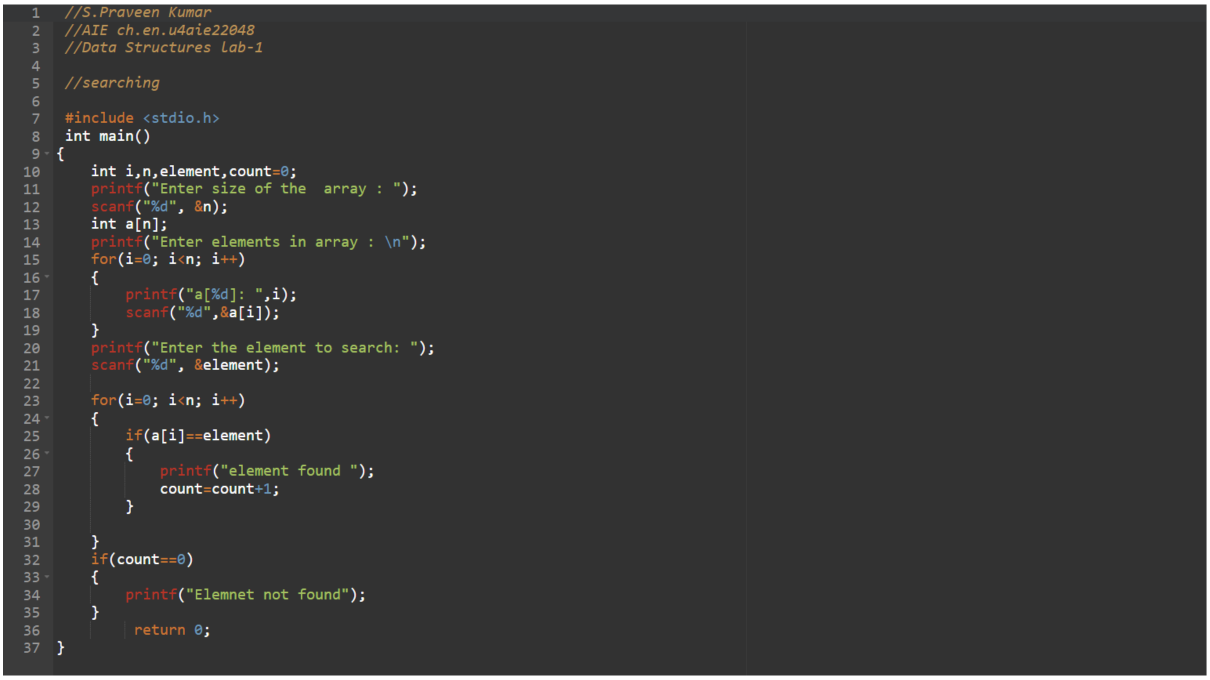
```
1.Sorting  at ascending
2.Sorting at descending
Enter a number: 1
enter size of the element in array:
4
enter the elemnet in array:
a[0]: 1
a[1]: 23
a[2]: 45
a[3]: 678
After Sortingin ascending--------:
1
23
45
678


...Program finished with exit code 0
Press ENTER to exit console.
```



```
1.Sorting  at ascending
2.Sorting at descending
Enter a number: 2
enter size of the element in array:
4
enter the elemnet in array:
a[0]: 1
a[1]: 23
a[2]: 454
a[3]: 63
After Sortingin descending--------:
454
63
23
1


...Program finished with exit code 0
Press ENTER to exit console.
```

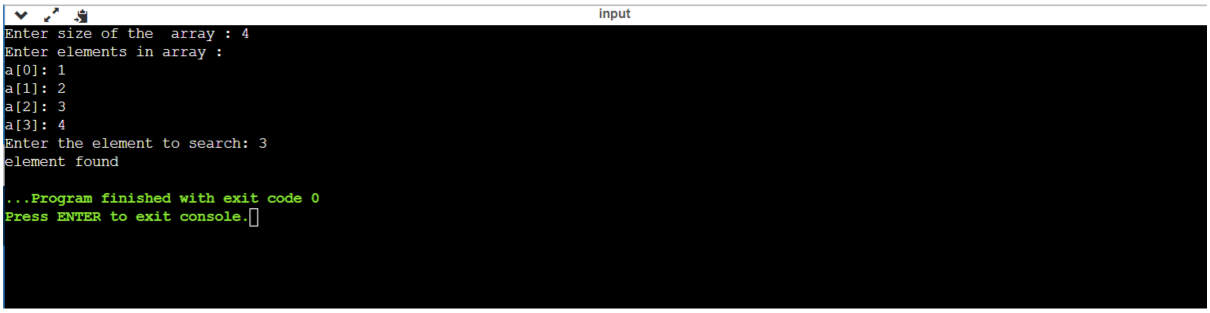
Searching

Program:



```c
//S.Praveen Kumar
//AIE ch.en.u4aie22048
//Data Structures lab-1

//searching

#include <stdio.h>
int main()
{
    int i,n,element,count=0;
    printf("Enter size of the  array : ");
    scanf("%d", &n);
    int a[n];
    printf("Enter elements in array : \n");
    for(i=0; i<n; i++)
    {
        printf("a[%d]: ",i);
        scanf("%d",&a[i]);
    }
    printf("Enter the element to search: ");
    scanf("%d", &element);

    for(i=0; i<n; i++)
    {
        if(a[i]==element)
        {
            printf("element found ");
            count=count+1;
        }

    }
    if(count==0)
    {
        printf("Elemnet not found");
    }
        return 0;
}
```

Output:



```
Enter size of the  array : 4
Enter elements in array :
a[0]: 1
a[1]: 2
a[2]: 3
a[3]: 4
Enter the element to search: 3
element found

...Program finished with exit code 0
Press ENTER to exit console.
```

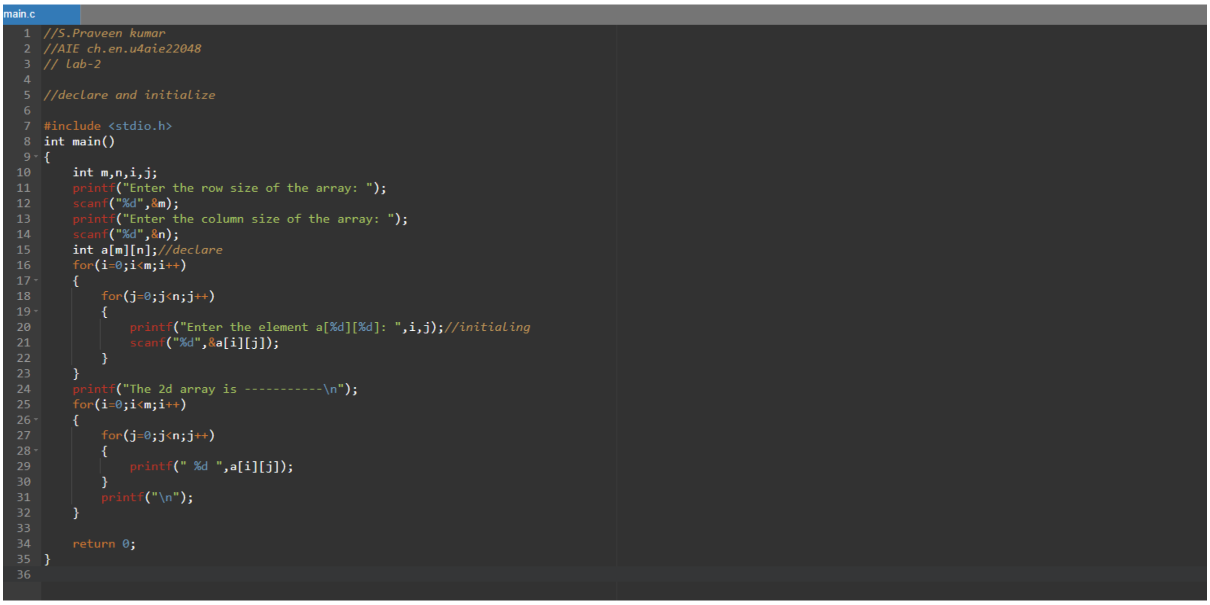
DATA STRUCTURE – 1

LAB-2

S.Praveen kumar
ch.en.u4aie22048

2-d array declare and initialize

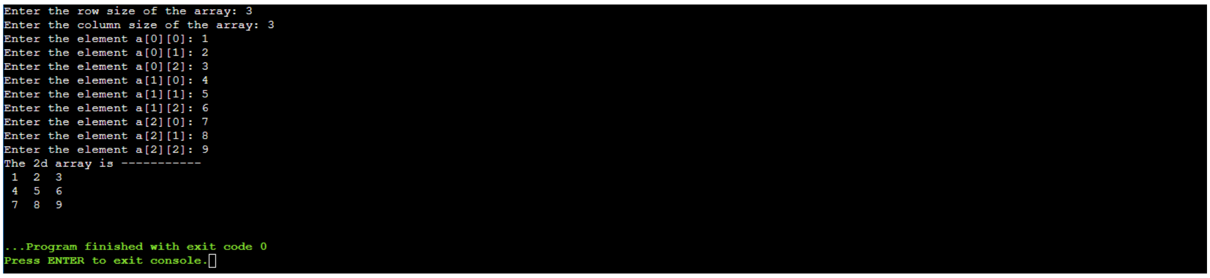
Program:



main.c

```c
//S.Praveen kumar
//AIE ch.en.u4aie22048
// lab-2

//declare and initialize

#include <stdio.h>
int main()
{
    int m,n,i,j;
    printf("Enter the row size of the array: ");
    scanf("%d",&m);
    printf("Enter the column size of the array: ");
    scanf("%d",&n);
    int a[m][n];//declare
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("Enter the element a[%d][%d]: ",i,j);//initialing
            scanf("%d",&a[i][j]);
        }
    }
    printf("The 2d array is -----------\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf(" %d ",a[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

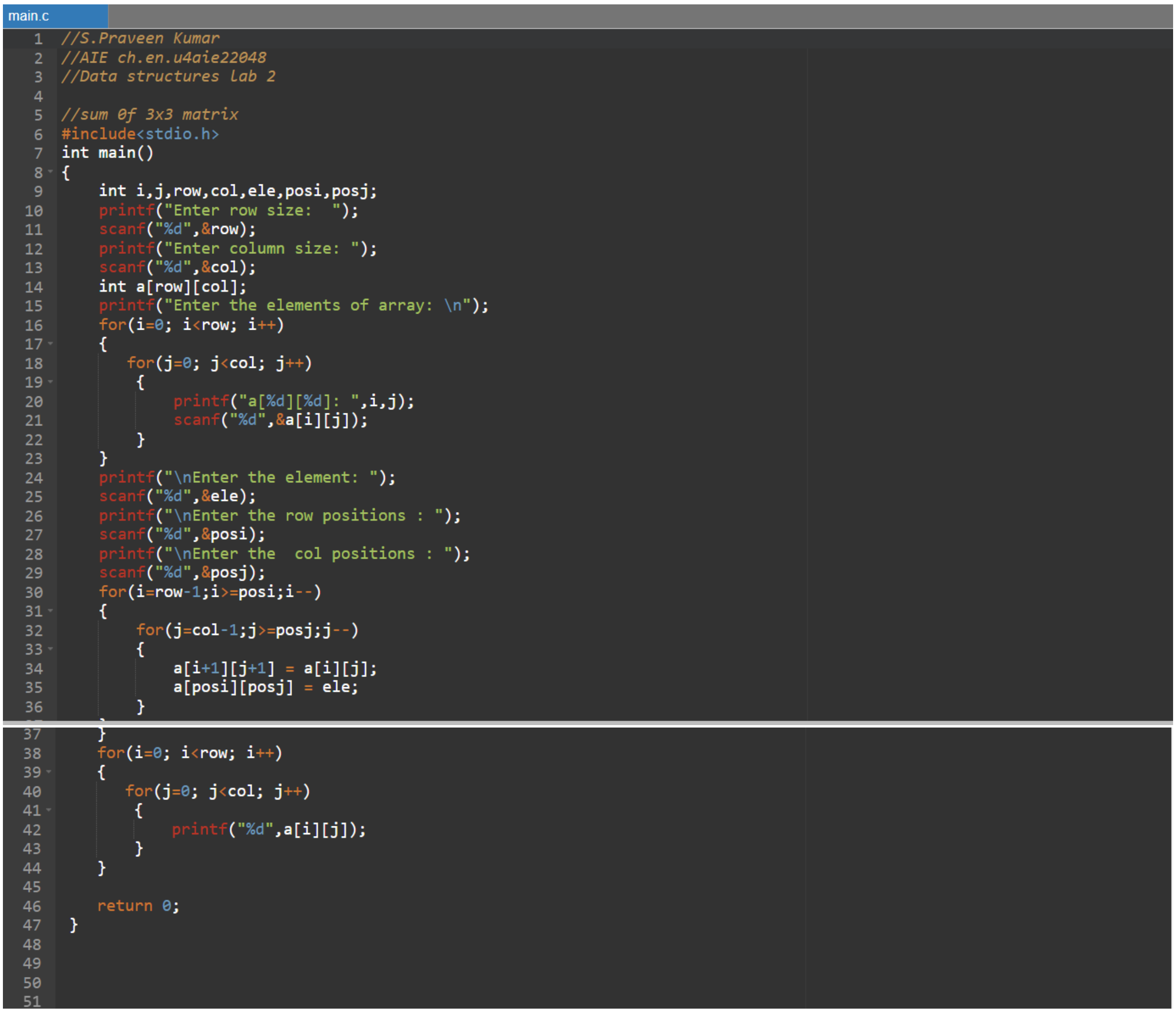
Output:



```
Enter the row size of the array: 3
Enter the column size of the array: 3
Enter the element a[0][0]: 1
Enter the element a[0][1]: 2
Enter the element a[0][2]: 3
Enter the element a[1][0]: 4
Enter the element a[1][1]: 5
Enter the element a[1][2]: 6
Enter the element a[2][0]: 7
Enter the element a[2][1]: 8
Enter the element a[2][2]: 9
The 2d array is -----------
 1  2  3
 4  5  6
 7  8  9


...Program finished with exit code 0
Press ENTER to exit console.
```

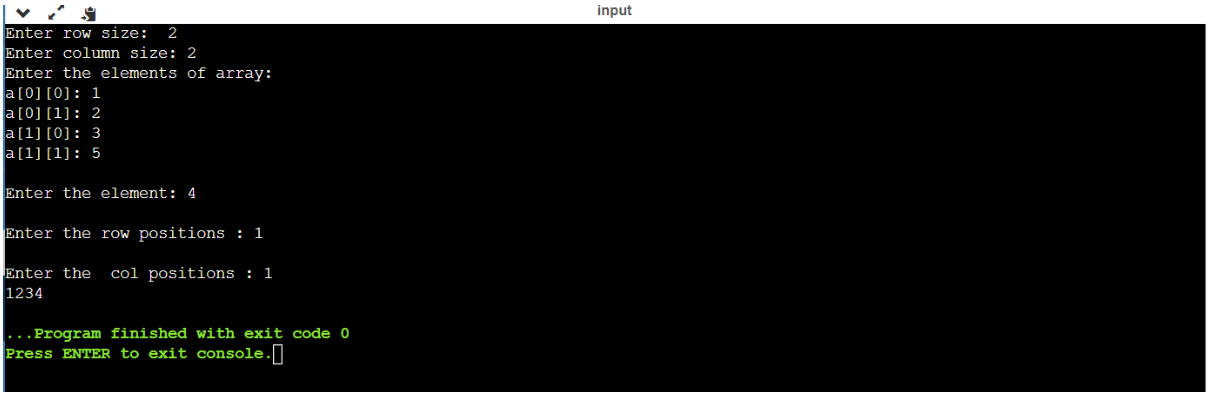
INSERTING IN 2D ARRAY

Program



```c
//S.Praveen Kumar
//AIE ch.en.u4aie22048
//Data structures lab 2

//sum 0f 3x3 matrix
#include<stdio.h>
int main()
{
    int i,j,row,col,ele,posi,posj;
    printf("Enter row size:  ");
    scanf("%d",&row);
    printf("Enter column size: ");
    scanf("%d",&col);
    int a[row][col];
    printf("Enter the elements of array: \n");
    for(i=0; i<row; i++)
    {
       for(j=0; j<col; j++)
        {
            printf("a[%d][%d]: ",i,j);
            scanf("%d",&a[i][j]);
        }
    }
    printf("\nEnter the element: ");
    scanf("%d",&ele);
    printf("\nEnter the row positions : ");
    scanf("%d",&posi);
    printf("\nEnter the  col positions : ");
    scanf("%d",&posj);
    for(i=row-1;i>=posi;i--)
    {
        for(j=col-1;j>=posj;j--)
        {
            a[i+1][j+1] = a[i][j];
            a[posi][posj] = ele;
        }
    }
    for(i=0; i<row; i++)
    {
       for(j=0; j<col; j++)
        {
            printf("%d",a[i][j]);
        }
    }

    return 0;
 }
```

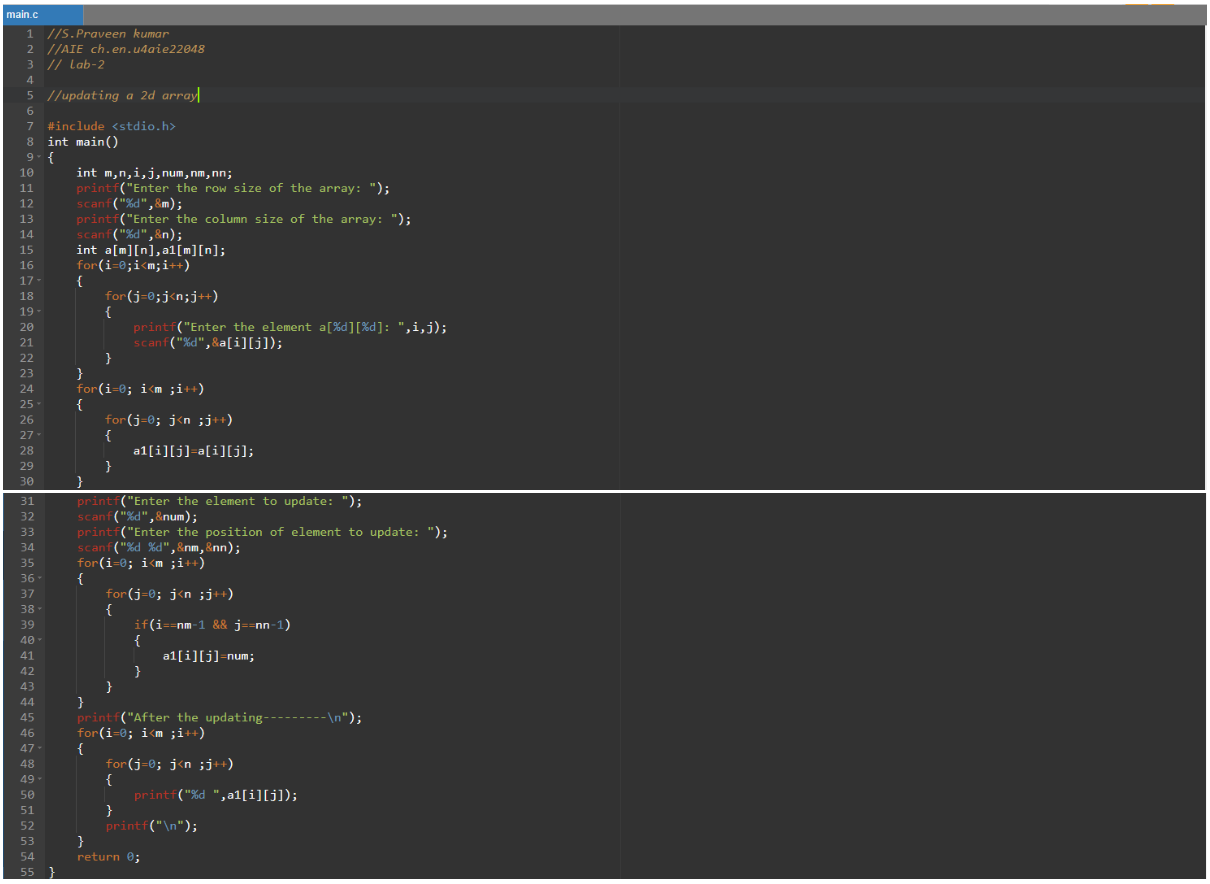
Output



```
Enter row size:  2
Enter column size: 2
Enter the elements of array:
a[0][0]: 1
a[0][1]: 2
a[1][0]: 3
a[1][1]: 5

Enter the element: 4

Enter the row positions : 1

Enter the  col positions : 1
1234

...Program finished with exit code 0
Press ENTER to exit console.
```

3.Updating in 2-d

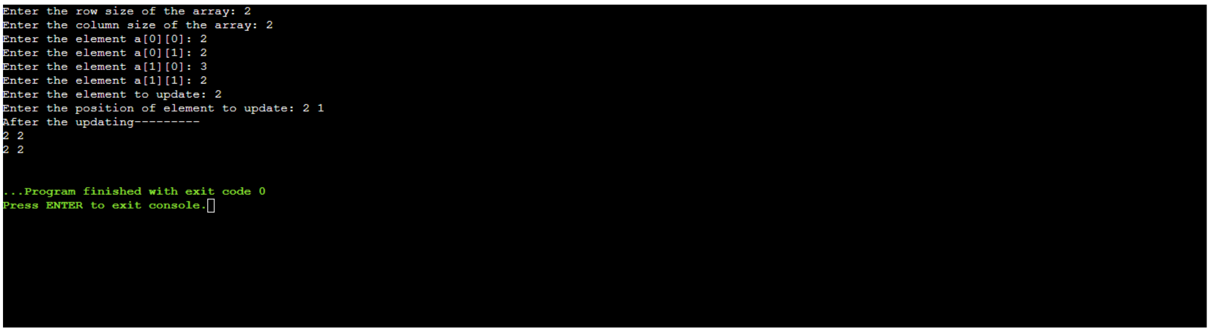
Program



main.c

```c
//S.Praveen kumar
//AIE ch.en.u4aie22048
// lab-2

//updating a 2d array

#include <stdio.h>
int main()
{
    int m,n,i,j,num,nm,nn;
    printf("Enter the row size of the array: ");
    scanf("%d",&m);
    printf("Enter the column size of the array: ");
    scanf("%d",&n);
    int a[m][n],a1[m][n];
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("Enter the element a[%d][%d]: ",i,j);
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0; i<m ;i++)
    {
        for(j=0; j<n ;j++)
        {
            a1[i][j]=a[i][j];
        }
    }
    printf("Enter the element to update: ");
    scanf("%d",&num);
    printf("Enter the position of element to update: ");
    scanf("%d %d",&nm,&nn);
    for(i=0; i<m ;i++)
    {
        for(j=0; j<n ;j++)
        {
            if(i==nm-1 && j==nn-1)
            {
                a1[i][j]=num;
            }
        }
    }
    printf("After the updating---------\n");
    for(i=0; i<m ;i++)
    {
        for(j=0; j<n ;j++)
        {
            printf("%d ",a1[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

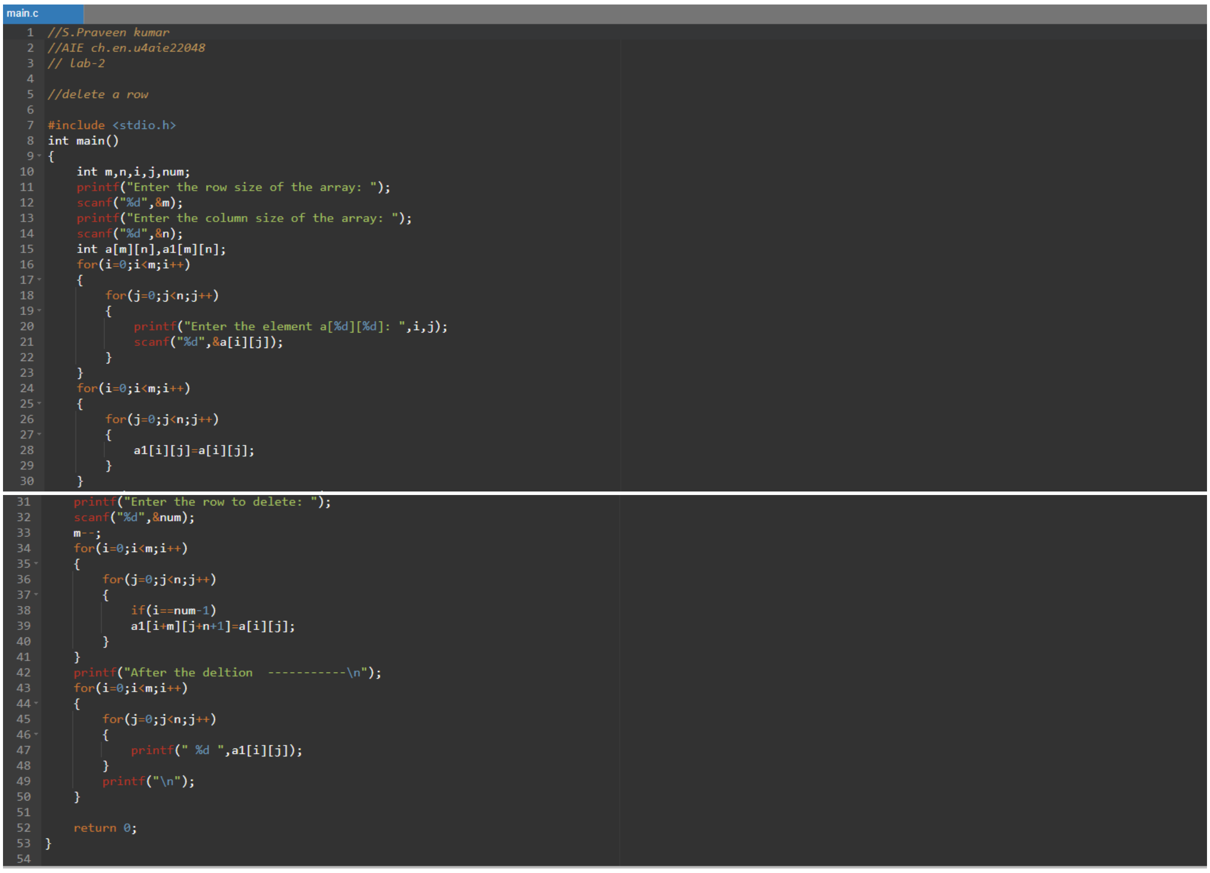
Output:



```
Enter the row size of the array: 2
Enter the column size of the array: 2
Enter the element a[0][0]: 2
Enter the element a[0][1]: 2
Enter the element a[1][0]: 3
Enter the element a[1][1]: 2
Enter the element to update: 2
Enter the position of element to update: 2 1
After the updating---------
2 2
2 2


...Program finished with exit code 0
Press ENTER to exit console.
```

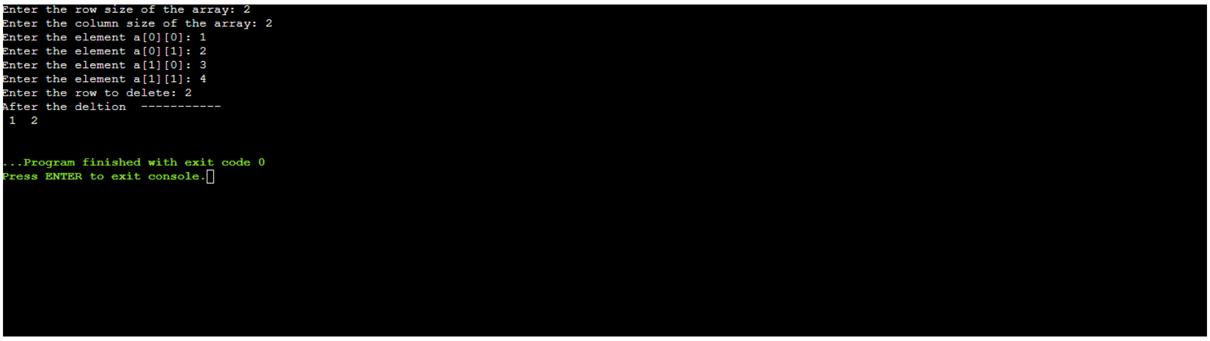
Deletion at row

Program:



```c
//S.Praveen kumar
//AIE ch.en.u4aie22048
// lab-2

//delete a row

#include <stdio.h>
int main()
{
    int m,n,i,j,num;
    printf("Enter the row size of the array: ");
    scanf("%d",&m);
    printf("Enter the column size of the array: ");
    scanf("%d",&n);
    int a[m][n],a1[m][n];
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf("Enter the element a[%d][%d]: ",i,j);
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            a1[i][j]=a[i][j];
        }
    }
    printf("Enter the row to delete: ");
    scanf("%d",&num);
    m--;
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            if(i==num-1)
            a1[i+m][j+n+1]=a[i][j];
        }
    }
    printf("After the deltion  -----------\n");
    for(i=0;i<m;i++)
    {
        for(j=0;j<n;j++)
        {
            printf(" %d ",a1[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

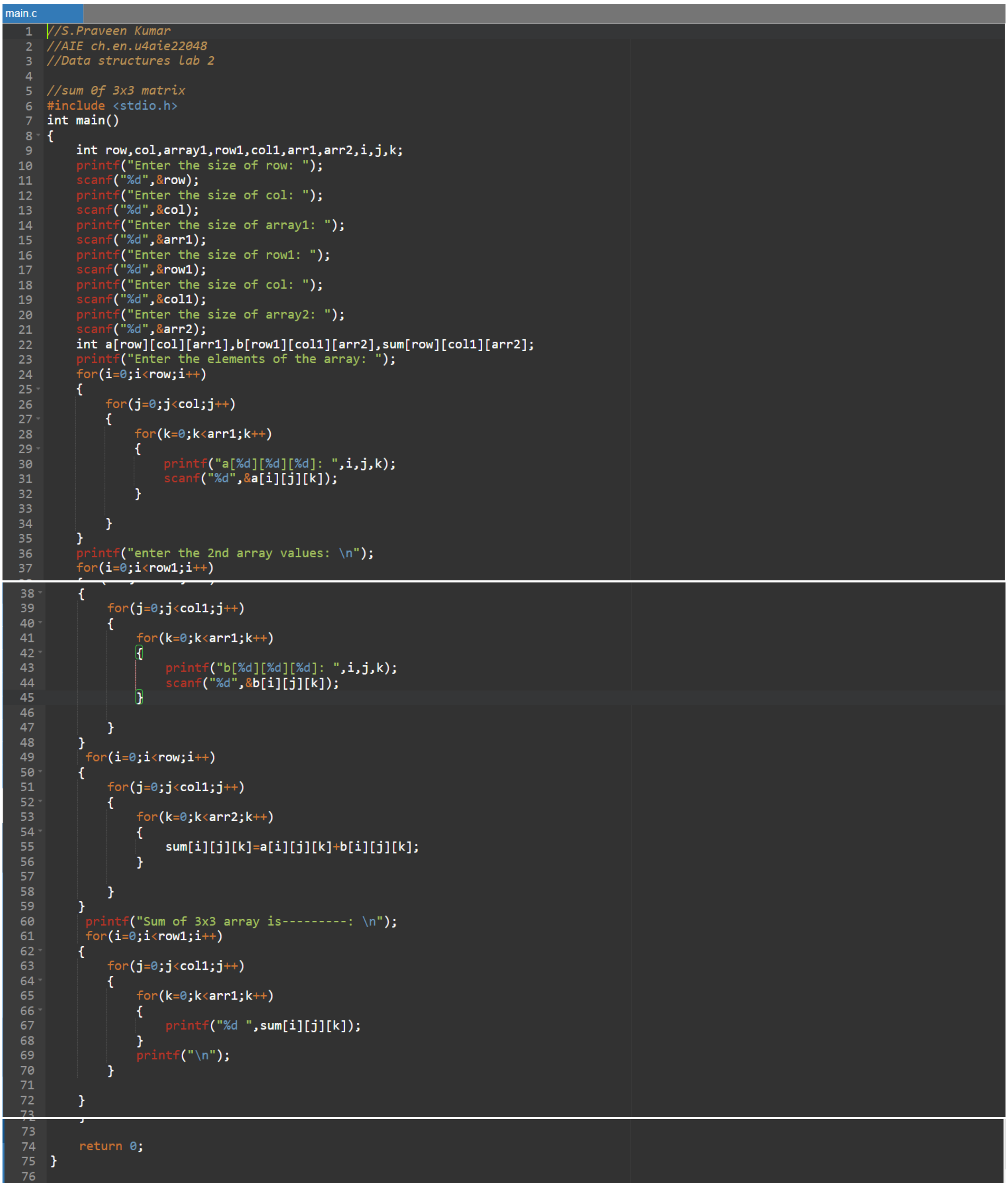
Output:



```
Enter the row size of the array: 2
Enter the column size of the array: 2
Enter the element a[0][0]: 1
Enter the element a[0][1]: 2
Enter the element a[1][0]: 3
Enter the element a[1][1]: 4
Enter the row to delete: 2
After the deltion  -----------
 1  2


...Program finished with exit code 0
Press ENTER to exit console.
```

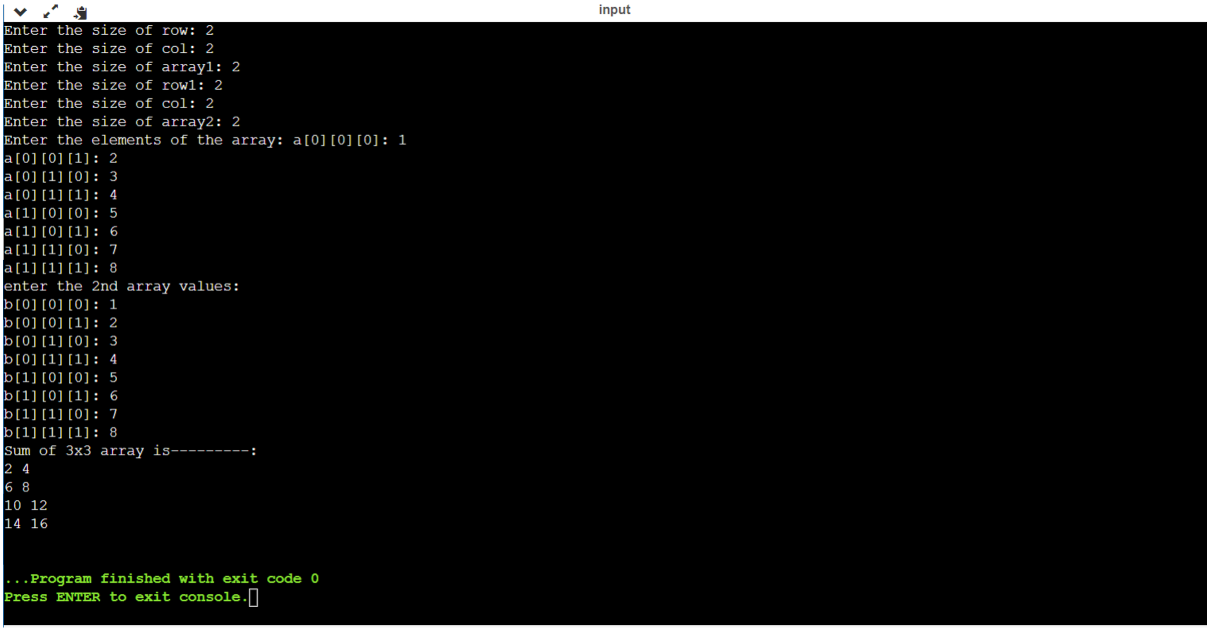
Sum of 3x3 matrix

Program



```c
//S.Praveen Kumar
//AIE ch.en.u4aie22048
//Data structures lab 2

//sum 0f 3x3 matrix
#include <stdio.h>
int main()
{
    int row,col,array1,row1,col1,arr1,arr2,i,j,k;
    printf("Enter the size of row: ");
    scanf("%d",&row);
    printf("Enter the size of col: ");
    scanf("%d",&col);
    printf("Enter the size of array1: ");
    scanf("%d",&arr1);
    printf("Enter the size of row1: ");
    scanf("%d",&row1);
    printf("Enter the size of col: ");
    scanf("%d",&col1);
    printf("Enter the size of array2: ");
    scanf("%d",&arr2);
    int a[row][col][arr1],b[row1][col1][arr2],sum[row][col1][arr2];
    printf("Enter the elements of the array: ");
    for(i=0;i<row;i++)
    {
        for(j=0;j<col;j++)
        {
            for(k=0;k<arr1;k++)
            {
                printf("a[%d][%d][%d]: ",i,j,k);
                scanf("%d",&a[i][j][k]);
            }

        }
    }
    printf("enter the 2nd array values: \n");
    for(i=0;i<row1;i++)
    {
        for(j=0;j<col1;j++)
        {
            for(k=0;k<arr1;k++)
            {
                printf("b[%d][%d][%d]: ",i,j,k);
                scanf("%d",&b[i][j][k]);
            }

        }
    }
     for(i=0;i<row;i++)
    {
        for(j=0;j<col1;j++)
        {
            for(k=0;k<arr2;k++)
            {
                sum[i][j][k]=a[i][j][k]+b[i][j][k];
            }

        }
    }
     printf("Sum of 3x3 array is---------: \n");
     for(i=0;i<row1;i++)
    {
        for(j=0;j<col1;j++)
        {
            for(k=0;k<arr1;k++)
            {
                printf("%d ",sum[i][j][k]);
            }
            printf("\n");
        }

    }

    return 0;
}
```

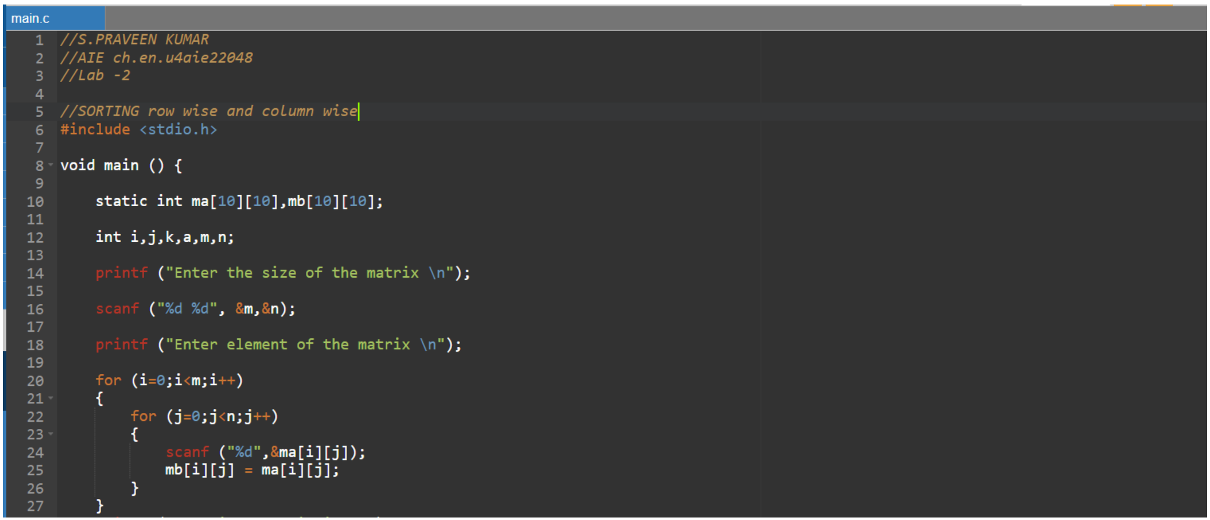
Output:



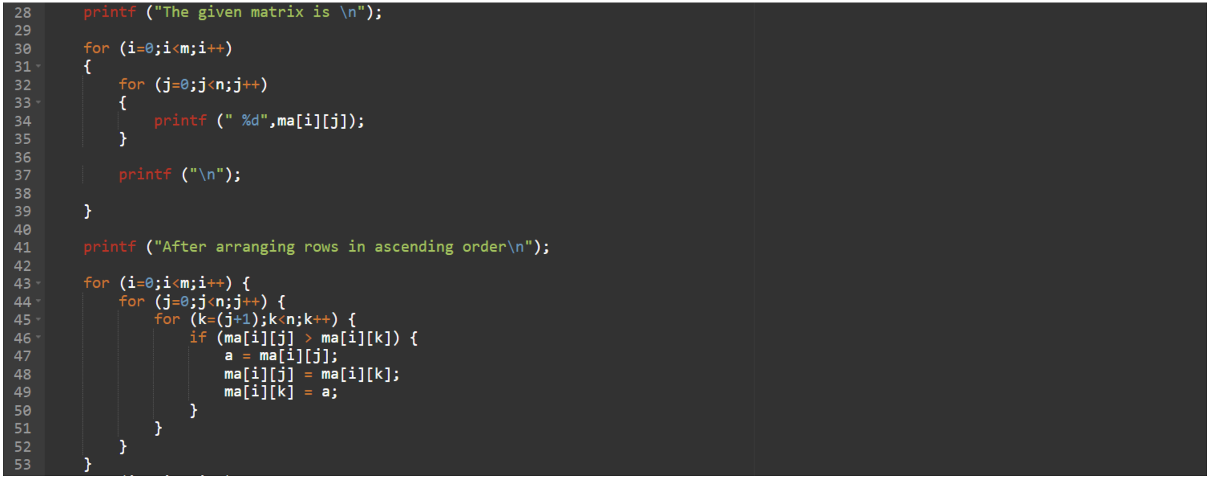
```
Enter the size of row: 2
Enter the size of col: 2
Enter the size of array1: 2
Enter the size of row1: 2
Enter the size of col: 2
Enter the size of array2: 2
Enter the elements of the array: a[0][0][0]: 1
a[0][0][1]: 2
a[0][1][0]: 3
a[0][1][1]: 4
a[1][0][0]: 5
a[1][0][1]: 6
a[1][1][0]: 7
a[1][1][1]: 8
enter the 2nd array values:
b[0][0][0]: 1
b[0][0][1]: 2
b[0][1][0]: 3
b[0][1][1]: 4
b[1][0][0]: 5
b[1][0][1]: 6
b[1][1][0]: 7
b[1][1][1]: 8
Sum of 3x3 array is---------:
2 4
6 8
10 12
14 16


...Program finished with exit code 0
Press ENTER to exit console.
```

SORTING IN 2D ARRAY

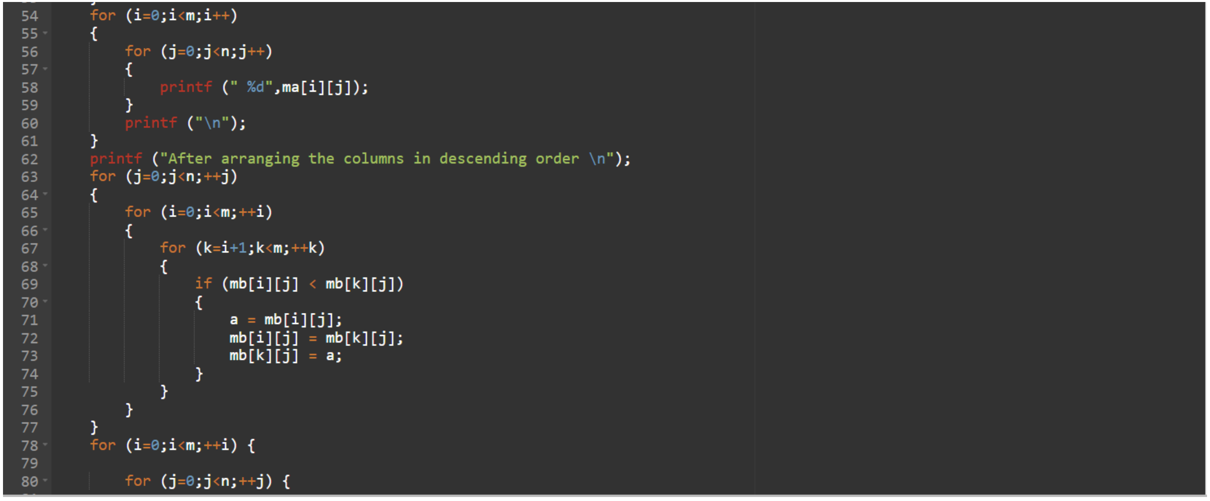
PROGRAM:



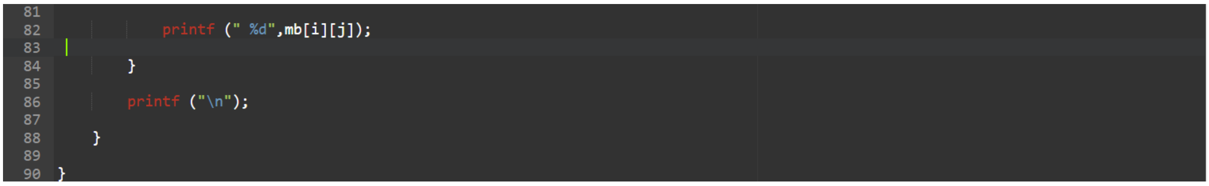
```c
//S.PRAVEEN KUMAR
//AIE ch.en.u4aie22048
//Lab -2

//SORTING row wise and column wise
#include <stdio.h>

void main () {

    static int ma[10][10],mb[10][10];

    int i,j,k,a,m,n;

    printf ("Enter the size of the matrix \n");

    scanf ("%d %d", &m,&n);

    printf ("Enter element of the matrix \n");

    for (i=0;i<m;i++)
    {
        for (j=0;j<n;j++)
        {
            scanf ("%d",&ma[i][j]);
            mb[i][j] = ma[i][j];
        }
    }
```



```c
    printf ("The given matrix is \n");

    for (i=0;i<m;i++)
    {
        for (j=0;j<n;j++)
        {
            printf (" %d",ma[i][j]);
        }

        printf ("\n");

    }

    printf ("After arranging rows in ascending order\n");

    for (i=0;i<m;i++) {
        for (j=0;j<n;j++) {
            for (k=(j+1);k<n;k++) {
                if (ma[i][j] > ma[i][k]) {
                    a = ma[i][j];
                    ma[i][j] = ma[i][k];
                    ma[i][k] = a;
                }
            }
        }
    }
```

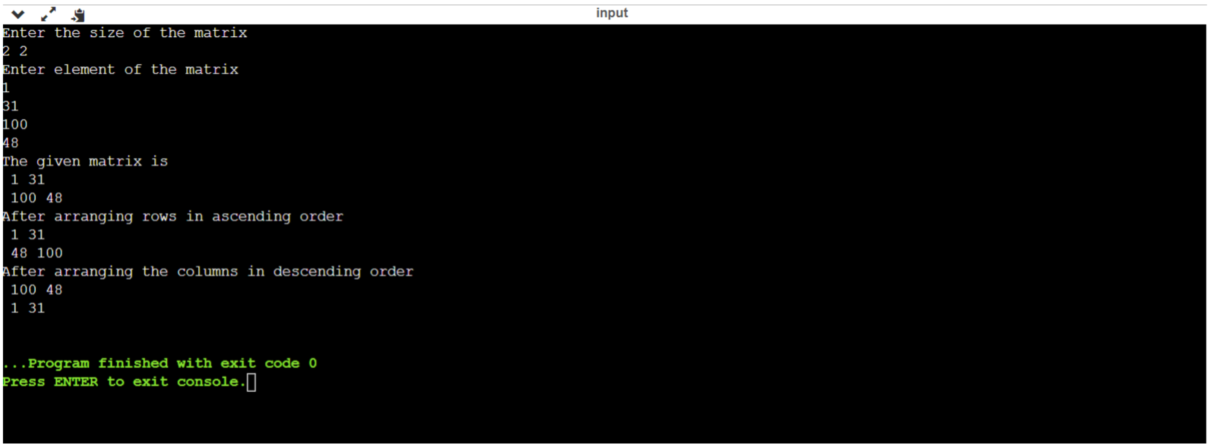


```c
    for (i=0;i<m;i++)
    {
        for (j=0;j<n;j++)
        {
            printf (" %d",ma[i][j]);
        }
        printf ("\n");
    }
    printf ("After arranging the columns in descending order \n");
    for (j=0;j<n;++j)
    {
        for (i=0;i<m;++i)
        {
            for (k=i+1;k<m;++k)
            {
                if (mb[i][j] < mb[k][j])
                {
                    a = mb[i][j];
                    mb[i][j] = mb[k][j];
                    mb[k][j] = a;
                }
            }
        }
    }
    for (i=0;i<m;++i) {

        for (j=0;j<n;++j) {
```



```c

            printf (" %d",mb[i][j]);

        }

        printf ("\n");

    }

}
```

OUTPUT:



```
Enter the size of the matrix
2 2
Enter element of the matrix
1
31
100
48
The given matrix is
 1 31
 100 48
After arranging rows in ascending order
 1 31
 48 100
After arranging the columns in descending order
 100 48
 1 31


...Program finished with exit code 0
Press ENTER to exit console.
```

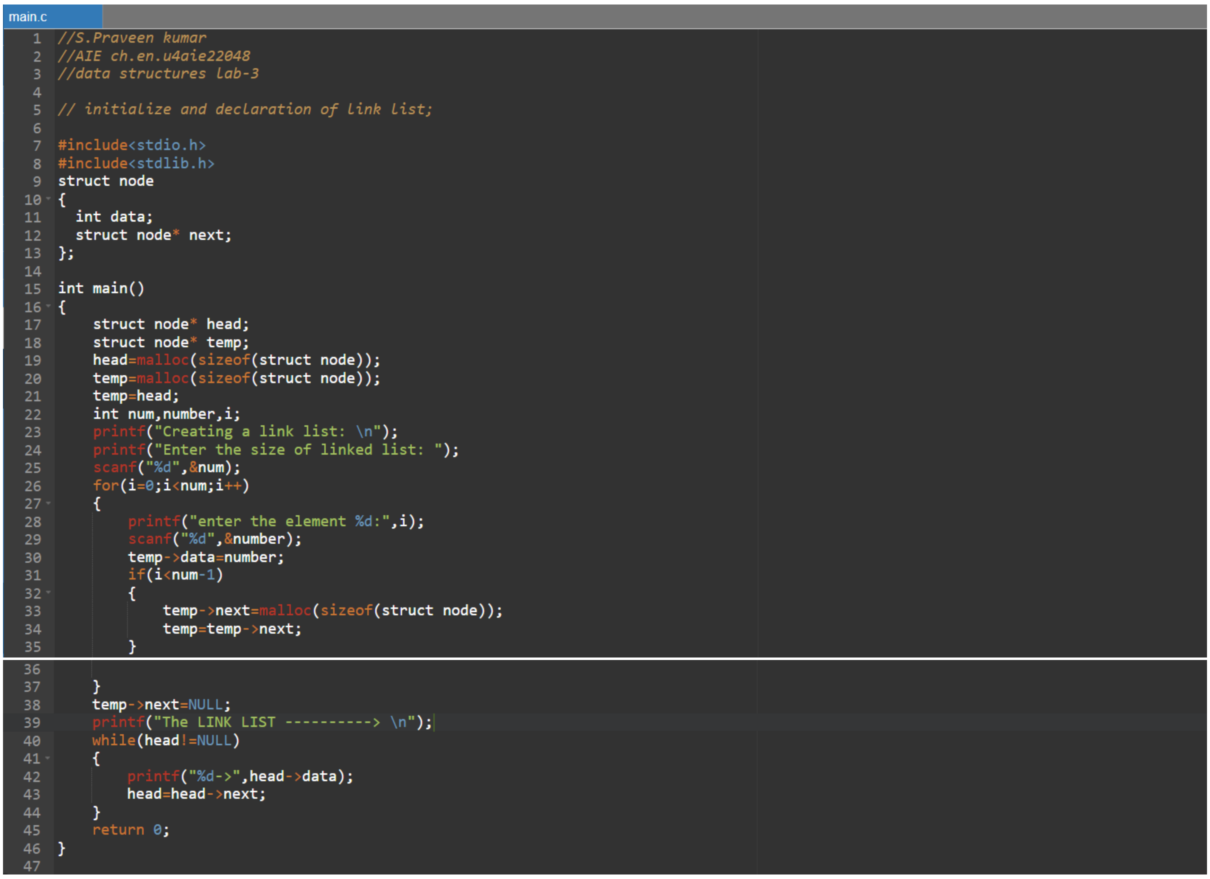
DATA STRUCTURE – 1

LAB-3

S.Praveen kumar
ch.en.u4aie22048

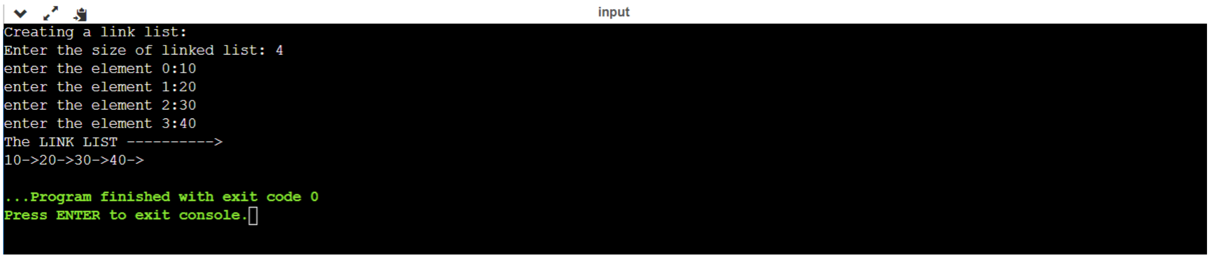
Initialize and declaration in link list:

Program:



```c
//S.Praveen kumar
//AIE ch.en.u4aie22048
//data structures lab-3

// initialize and declaration of link list;

#include<stdio.h>
#include<stdlib.h>
struct node
{
  int data;
  struct node* next;
};

int main()
{
    struct node* head;
    struct node* temp;
    head=malloc(sizeof(struct node));
    temp=malloc(sizeof(struct node));
    temp=head;
    int num,number,i;
    printf("Creating a link list: \n");
    printf("Enter the size of linked list: ");
    scanf("%d",&num);
    for(i=0;i<num;i++)
    {
        printf("enter the element %d:",i);
        scanf("%d",&number);
        temp->data=number;
        if(i<num-1)
        {
            temp->next=malloc(sizeof(struct node));
            temp=temp->next;
        }

    }
    temp->next=NULL;
    printf("The LINK LIST ----------> \n");
    while(head!=NULL)
    {
        printf("%d->",head->data);
        head=head->next;
    }
    return 0;
}

```

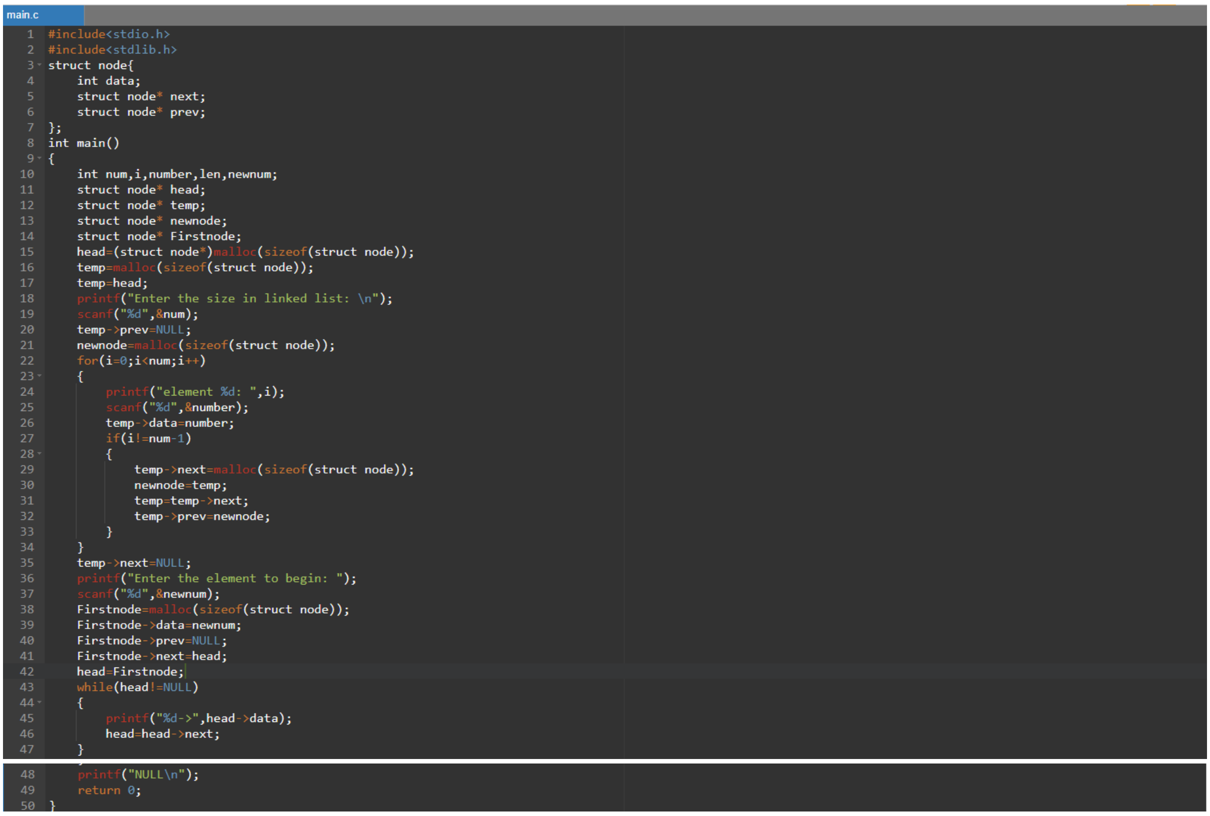
Output:



```
Creating a link list:
Enter the size of linked list: 4
enter the element 0:10
enter the element 1:20
enter the element 2:30
enter the element 3:40
The LINK LIST ---------->
10->20->30->40->

...Program finished with exit code 0
Press ENTER to exit console.
```

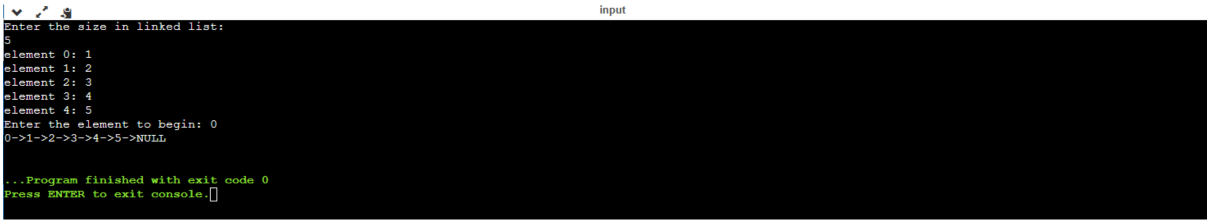
Insertion at beginning:

Program:



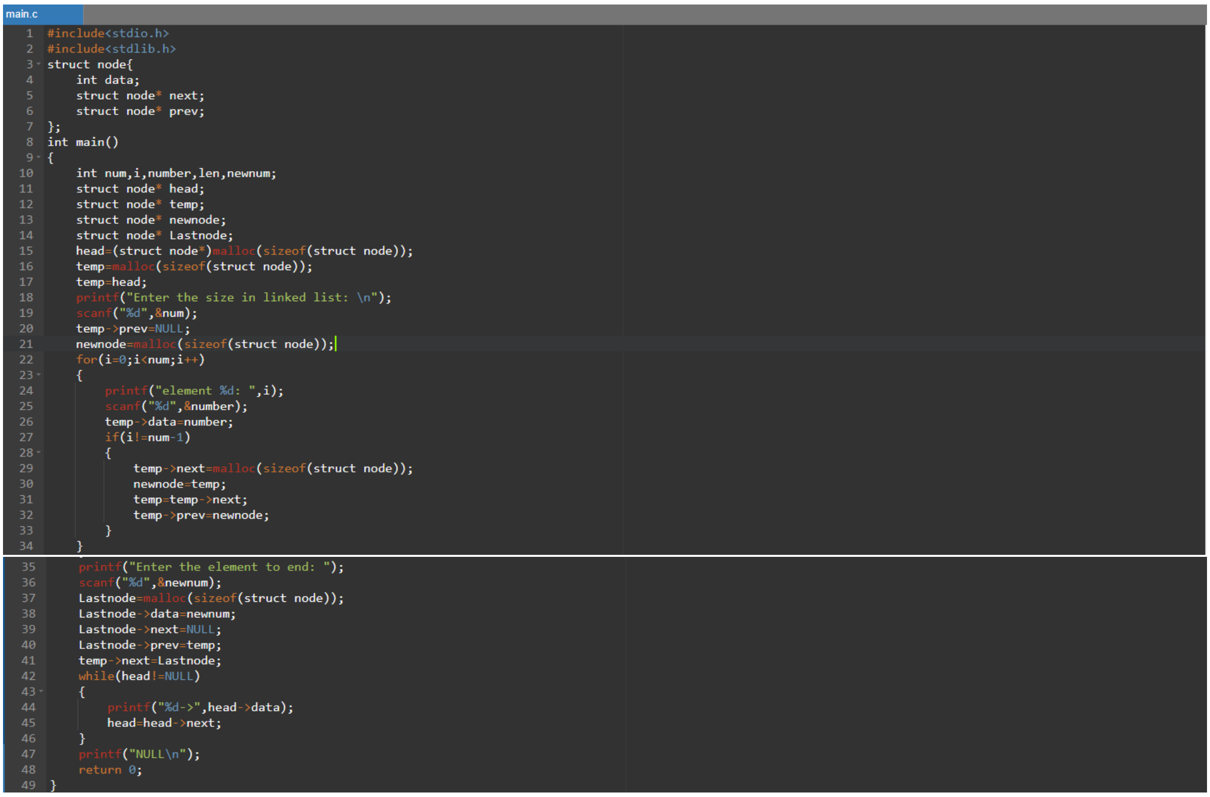
```c
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node* next;
    struct node* prev;
};
int main()
{
    int num,i,number,len,newnum;
    struct node* head;
    struct node* temp;
    struct node* newnode;
    struct node* Firstnode;
    head=(struct node*)malloc(sizeof(struct node));
    temp=malloc(sizeof(struct node));
    temp=head;
    printf("Enter the size in linked list: \n");
    scanf("%d",&num);
    temp->prev=NULL;
    newnode=malloc(sizeof(struct node));
    for(i=0;i<num;i++)
    {
        printf("element %d: ",i);
        scanf("%d",&number);
        temp->data=number;
        if(i!=num-1)
        {
            temp->next=malloc(sizeof(struct node));
            newnode=temp;
            temp=temp->next;
            temp->prev=newnode;
        }
    }
    temp->next=NULL;
    printf("Enter the element to begin: ");
    scanf("%d",&newnum);
    Firstnode=malloc(sizeof(struct node));
    Firstnode->data=newnum;
    Firstnode->prev=NULL;
    Firstnode->next=head;
    head=Firstnode;
    while(head!=NULL)
    {
        printf("%d->",head->data);
        head=head->next;
    }
    printf("NULL\n");
    return 0;
}
```

Output:



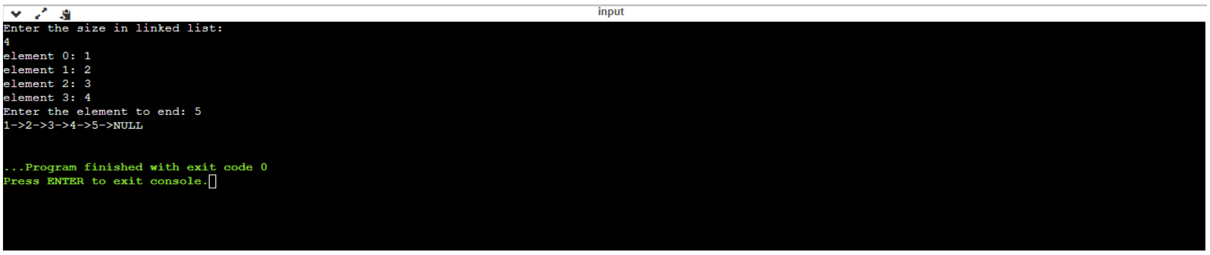
```
Enter the size in linked list:
5
element 0: 1
element 1: 2
element 2: 3
element 3: 4
element 4: 5
Enter the element to begin: 0
0->1->2->3->4->5->NULL


...Program finished with exit code 0
Press ENTER to exit console.
```

Insertion at end:



```c
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node* next;
    struct node* prev;
};
int main()
{
    int num,i,number,len,newnum;
    struct node* head;
    struct node* temp;
    struct node* newnode;
    struct node* Lastnode;
    head=(struct node*)malloc(sizeof(struct node));
    temp=malloc(sizeof(struct node));
    temp=head;
    printf("Enter the size in linked list: \n");
    scanf("%d",&num);
    temp->prev=NULL;
    newnode=malloc(sizeof(struct node));
    for(i=0;i<num;i++)
    {
        printf("element %d: ",i);
        scanf("%d",&number);
        temp->data=number;
        if(i!=num-1)
        {
            temp->next=malloc(sizeof(struct node));
            newnode=temp;
            temp=temp->next;
            temp->prev=newnode;
        }
    }
    printf("Enter the element to end: ");
    scanf("%d",&newnum);
    Lastnode=malloc(sizeof(struct node));
    Lastnode->data=newnum;
    Lastnode->next=NULL;
    Lastnode->prev=temp;
    temp->next=Lastnode;
    while(head!=NULL)
    {
        printf("%d->",head->data);
        head=head->next;
    }
    printf("NULL\n");
    return 0;
}
```

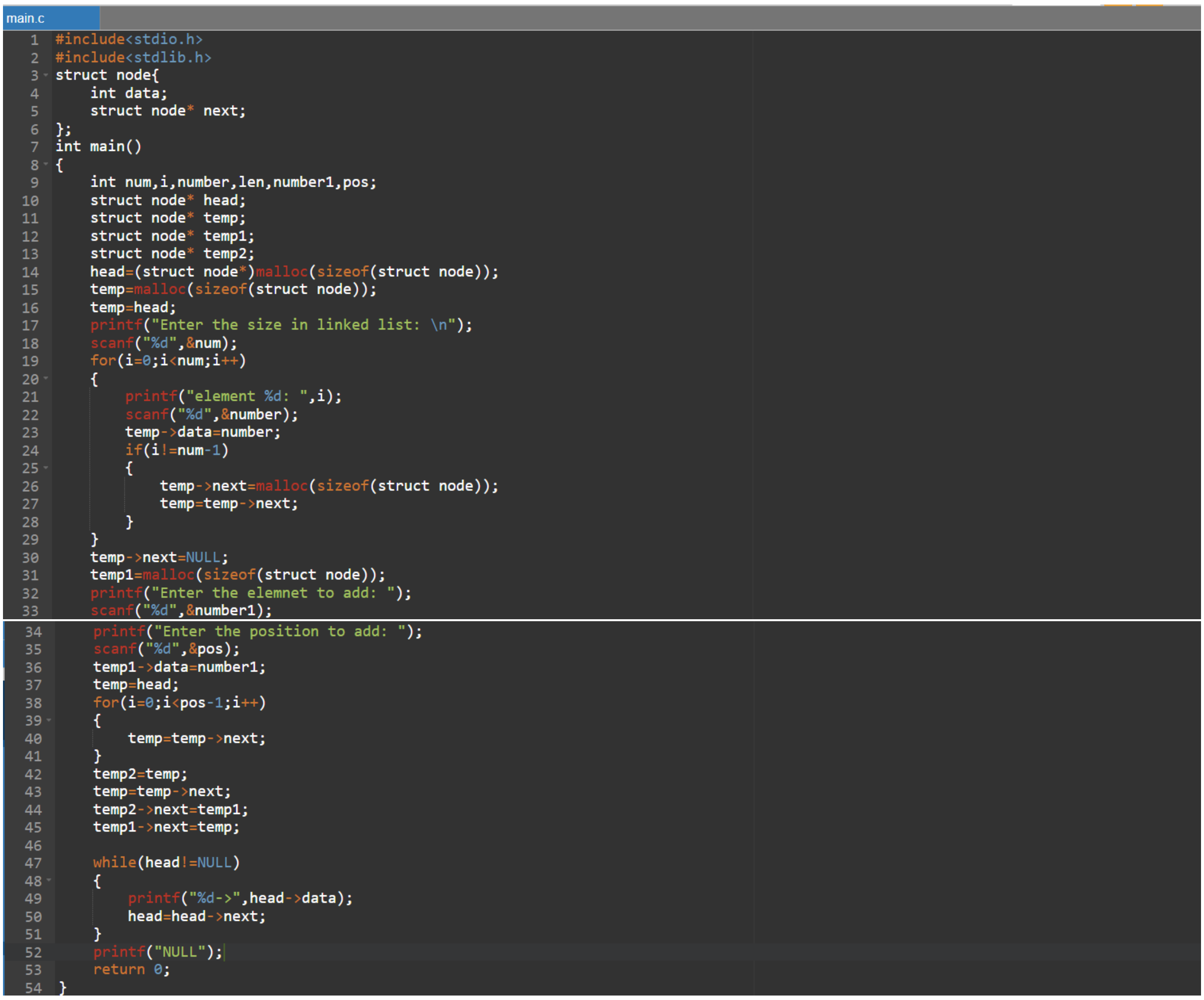
Output:



```
Enter the size in linked list:
4
element 0: 1
element 1: 2
element 2: 3
element 3: 4
Enter the element to end: 5
1->2->3->4->5->NULL


...Program finished with exit code 0
Press ENTER to exit console.
```

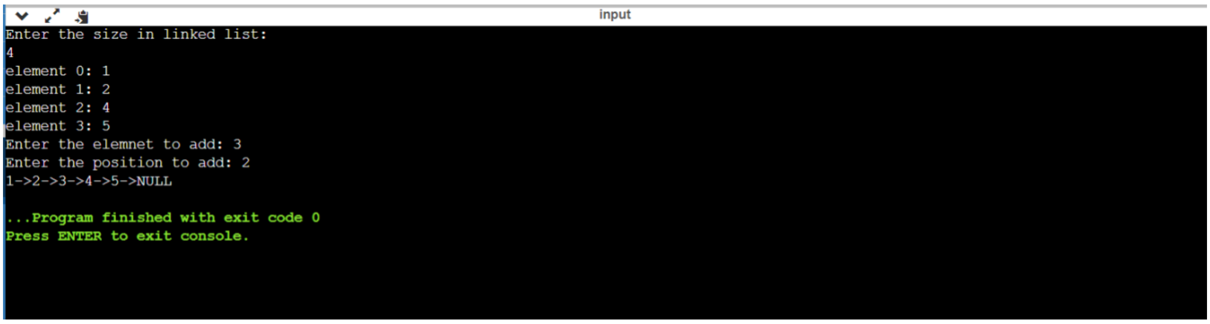
Inserting at specific position:

Program:



```c
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node* next;
};
int main()
{
    int num,i,number,len,number1,pos;
    struct node* head;
    struct node* temp;
    struct node* temp1;
    struct node* temp2;
    head=(struct node*)malloc(sizeof(struct node));
    temp=malloc(sizeof(struct node));
    temp=head;
    printf("Enter the size in linked list: \n");
    scanf("%d",&num);
    for(i=0;i<num;i++)
    {
        printf("element %d: ",i);
        scanf("%d",&number);
        temp->data=number;
        if(i!=num-1)
        {
            temp->next=malloc(sizeof(struct node));
            temp=temp->next;
        }
    }
    temp->next=NULL;
    temp1=malloc(sizeof(struct node));
    printf("Enter the elemnet to add: ");
    scanf("%d",&number1);
    printf("Enter the position to add: ");
    scanf("%d",&pos);
    temp1->data=number1;
    temp=head;
    for(i=0;i<pos-1;i++)
    {
        temp=temp->next;
    }
    temp2=temp;
    temp=temp->next;
    temp2->next=temp1;
    temp1->next=temp;

    while(head!=NULL)
    {
        printf("%d->",head->data);
        head=head->next;
    }
    printf("NULL");
    return 0;
}
```

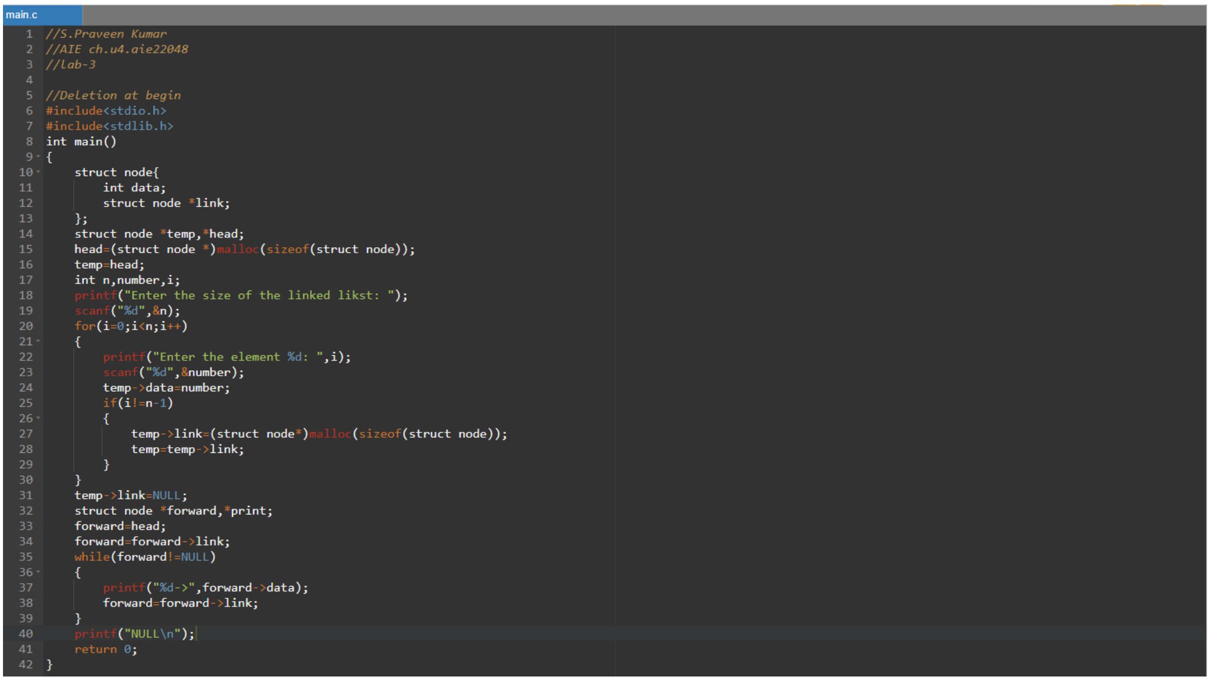
Output:



```
Enter the size in linked list:
4
element 0: 1
element 1: 2
element 2: 4
element 3: 5
Enter the elemnet to add: 3
Enter the position to add: 2
1->2->3->4->5->NULL

...Program finished with exit code 0
Press ENTER to exit console.
```

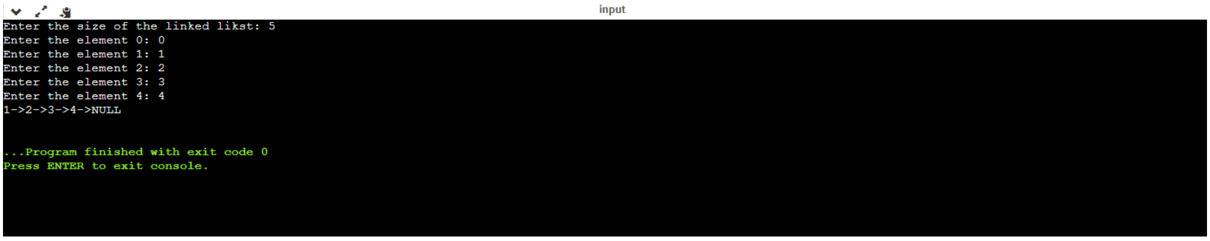
Deletion at beginning:

Program:



```c
//S.Praveen Kumar
//AIE ch.u4.aie22048
//lab-3

//Deletion at begin
#include<stdio.h>
#include<stdlib.h>
int main()
{
    struct node{
        int data;
        struct node *link;
    };
    struct node *temp,*head;
    head=(struct node *)malloc(sizeof(struct node));
    temp=head;
    int n,number,i;
    printf("Enter the size of the linked likst: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter the element %d: ",i);
        scanf("%d",&number);
        temp->data=number;
        if(i!=n-1)
        {
            temp->link=(struct node*)malloc(sizeof(struct node));
            temp=temp->link;
        }
    }
    temp->link=NULL;
    struct node *forward,*print;
    forward=head;
    forward=forward->link;
    while(forward!=NULL)
    {
        printf("%d->",forward->data);
        forward=forward->link;
    }
    printf("NULL\n");
    return 0;
}
```

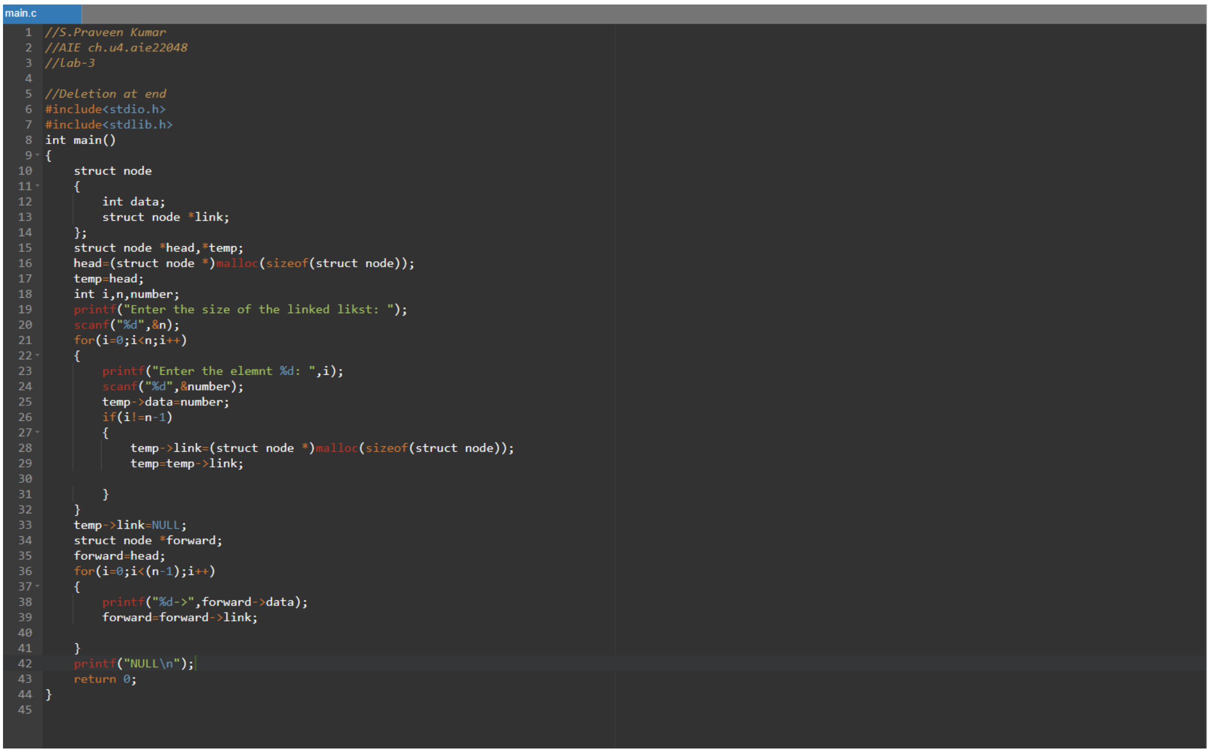
Output:



```
Enter the size of the linked likst: 5
Enter the element 0: 0
Enter the element 1: 1
Enter the element 2: 2
Enter the element 3: 3
Enter the element 4: 4
1->2->3->4->NULL


...Program finished with exit code 0
Press ENTER to exit console.
```

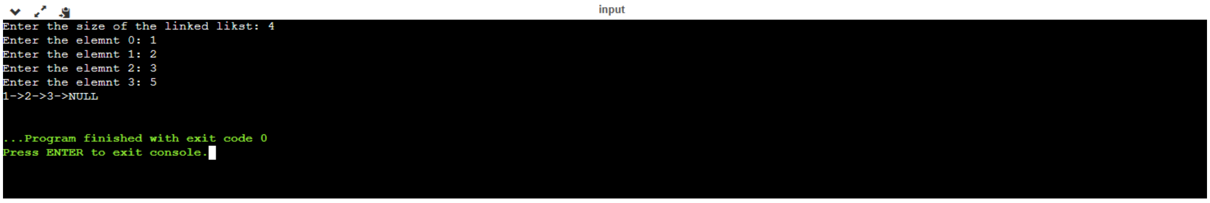
Deletion at end:

Program:



```c
//S.Praveen Kumar
//AIE ch.u4.aie22048
//lab-3

//Deletion at end
#include<stdio.h>
#include<stdlib.h>
int main()
{
    struct node
    {
        int data;
        struct node *link;
    };
    struct node *head,*temp;
    head=(struct node *)malloc(sizeof(struct node));
    temp=head;
    int i,n,number;
    printf("Enter the size of the linked likst: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter the elemnt %d: ",i);
        scanf("%d",&number);
        temp->data=number;
        if(i!=n-1)
        {
            temp->link=(struct node *)malloc(sizeof(struct node));
            temp=temp->link;

        }
    }
    temp->link=NULL;
    struct node *forward;
    forward=head;
    for(i=0;i<(n-1);i++)
    {
        printf("%d->",forward->data);
        forward=forward->link;

    }
    printf("NULL\n");
    return 0;
}
```

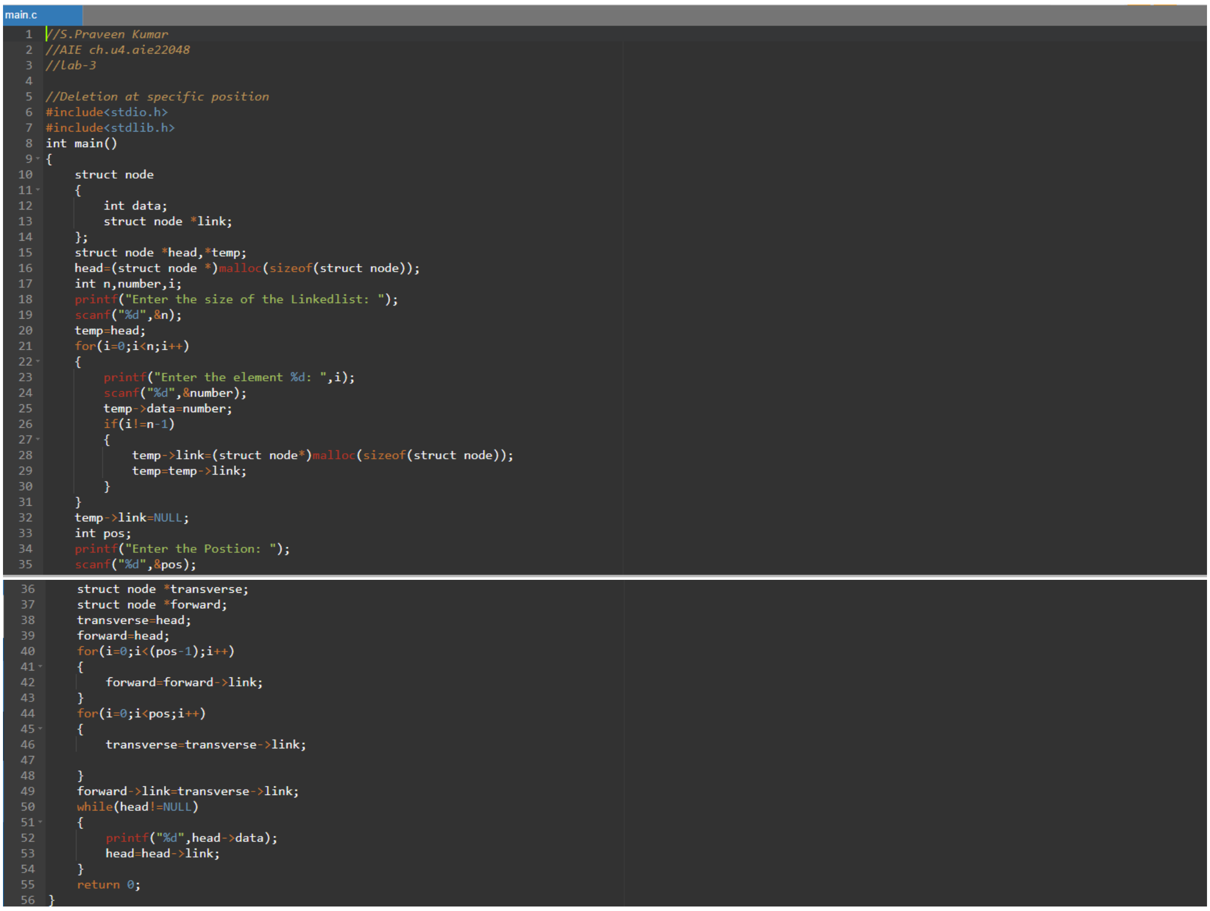
Output:



```
Enter the size of the linked likst: 4
Enter the elemnt 0: 1
Enter the elemnt 1: 2
Enter the elemnt 2: 3
Enter the elemnt 3: 5
1->2->3->NULL


...Program finished with exit code 0
Press ENTER to exit console.
```

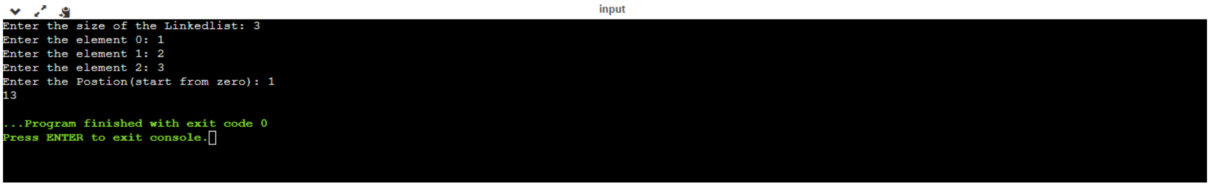
Deletion at specific position:

Program:



```
//S.Praveen Kumar
//AIE ch.u4.aie22048
//lab-3

//Deletion at specific position
#include<stdio.h>
#include<stdlib.h>
int main()
{
    struct node
    {
        int data;
        struct node *link;
    };
    struct node *head,*temp;
    head=(struct node *)malloc(sizeof(struct node));
    int n,number,i;
    printf("Enter the size of the Linkedlist: ");
    scanf("%d",&n);
    temp=head;
    for(i=0;i<n;i++)
    {
        printf("Enter the element %d: ",i);
        scanf("%d",&number);
        temp->data=number;
        if(i!=n-1)
        {
            temp->link=(struct node*)malloc(sizeof(struct node));
            temp=temp->link;
        }
    }
    temp->link=NULL;
    int pos;
    printf("Enter the Postion: ");
    scanf("%d",&pos);
    struct node *transverse;
    struct node *forward;
    transverse=head;
    forward=head;
    for(i=0;i<(pos-1);i++)
    {
        forward=forward->link;
    }
    for(i=0;i<pos;i++)
    {
        transverse=transverse->link;

    }
    forward->link=transverse->link;
    while(head!=NULL)
    {
        printf("%d",head->data);
        head=head->link;
    }
    return 0;
}
```

Output:



```
Enter the size of the Linkedlist: 3
Enter the element 0: 1
Enter the element 1: 2
Enter the element 2: 3
Enter the Postion(start from zero): 1
13

...Program finished with exit code 0
Press ENTER to exit console.
```

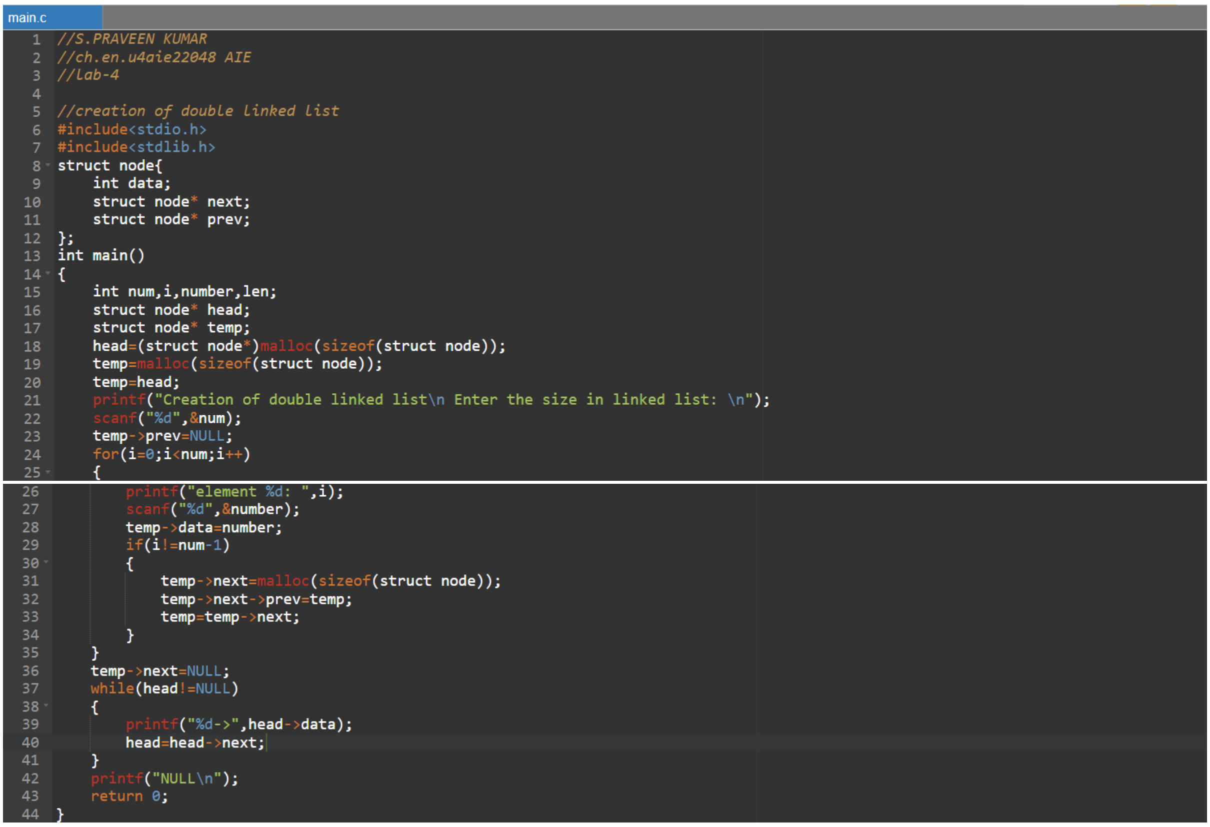
DATA STRUCTURE – 1

LAB-4

S.Praveen kumar
ch.en.u4aie22048

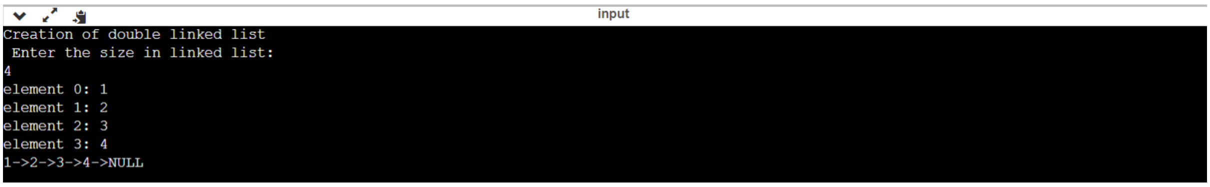
Initialize and declaration in Double link list:

Program:



```c
//S.PRAVEEN KUMAR
//ch.en.u4aie22048 AIE
//lab-4

//creation of double linked list
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node* next;
    struct node* prev;
};
int main()
{
    int num,i,number,len;
    struct node* head;
    struct node* temp;
    head=(struct node*)malloc(sizeof(struct node));
    temp=malloc(sizeof(struct node));
    temp=head;
    printf("Creation of double linked list\n Enter the size in linked list: \n");
    scanf("%d",&num);
    temp->prev=NULL;
    for(i=0;i<num;i++)
    {
        printf("element %d: ",i);
        scanf("%d",&number);
        temp->data=number;
        if(i!=num-1)
        {
            temp->next=malloc(sizeof(struct node));
            temp->next->prev=temp;
            temp=temp->next;
        }
    }
    temp->next=NULL;
    while(head!=NULL)
    {
        printf("%d->",head->data);
        head=head->next;
    }
    printf("NULL\n");
    return 0;
}
```

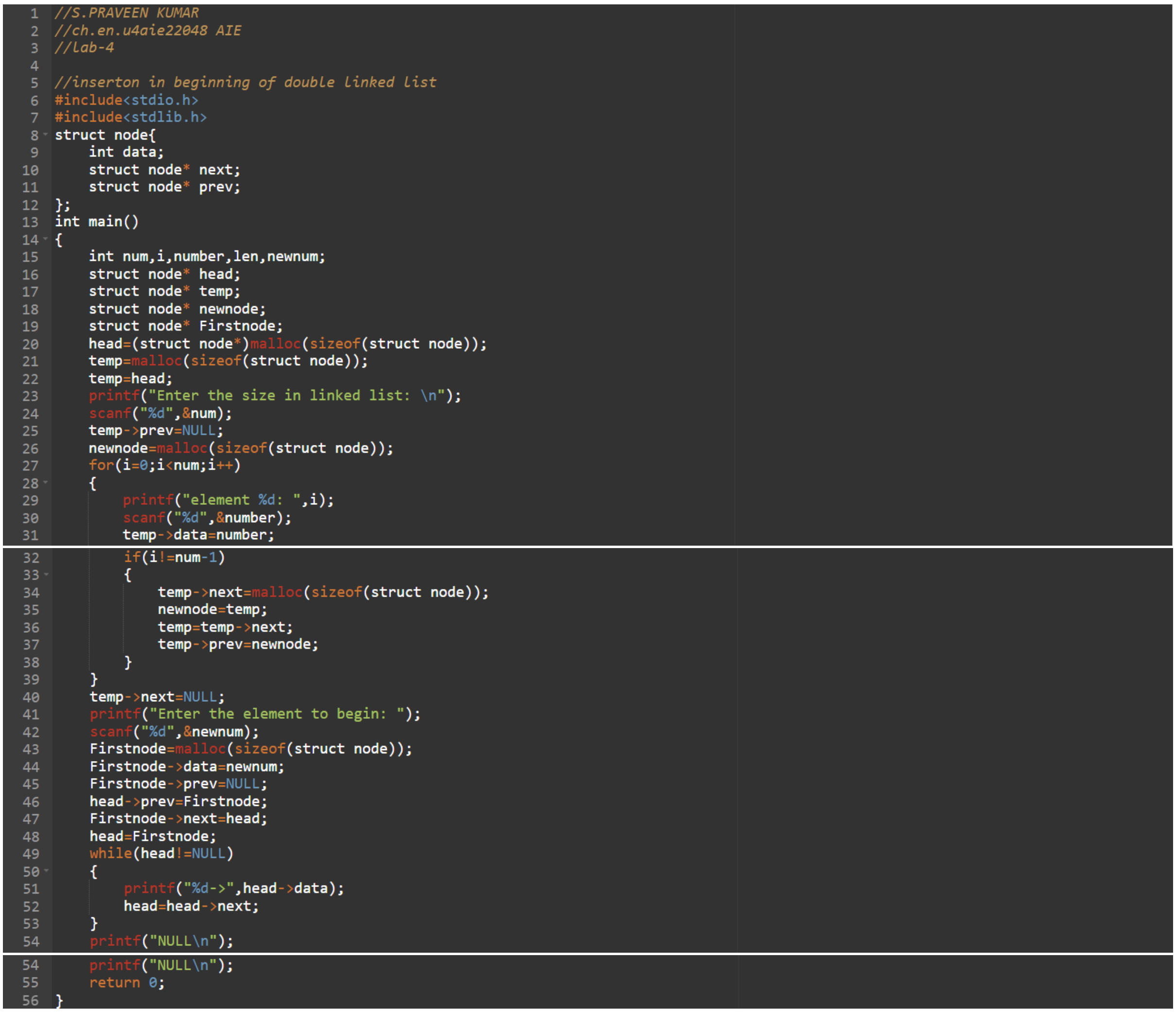
Output:



```
Creation of double linked list
 Enter the size in linked list:
4
element 0: 1
element 1: 2
element 2: 3
element 3: 4
1->2->3->4->NULL
```

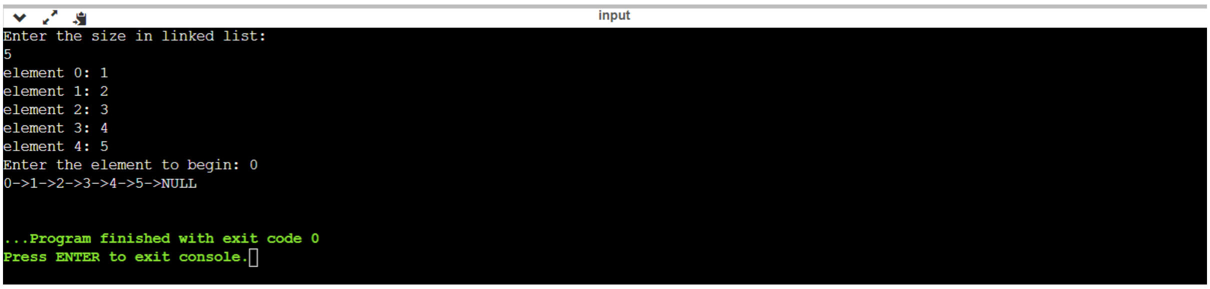
Insertion at beginning:

Program:



```c
//S.PRAVEEN KUMAR
//ch.en.u4aie22048 AIE
//lab-4

//inserton in beginning of double Linked List
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node* next;
    struct node* prev;
};
int main()
{
    int num,i,number,len,newnum;
    struct node* head;
    struct node* temp;
    struct node* newnode;
    struct node* Firstnode;
    head=(struct node*)malloc(sizeof(struct node));
    temp=malloc(sizeof(struct node));
    temp=head;
    printf("Enter the size in linked list: \n");
    scanf("%d",&num);
    temp->prev=NULL;
    newnode=malloc(sizeof(struct node));
    for(i=0;i<num;i++)
    {
        printf("element %d: ",i);
        scanf("%d",&number);
        temp->data=number;
        if(i!=num-1)
        {
            temp->next=malloc(sizeof(struct node));
            newnode=temp;
            temp=temp->next;
            temp->prev=newnode;
        }
    }
    temp->next=NULL;
    printf("Enter the element to begin: ");
    scanf("%d",&newnum);
    Firstnode=malloc(sizeof(struct node));
    Firstnode->data=newnum;
    Firstnode->prev=NULL;
    head->prev=Firstnode;
    Firstnode->next=head;
    head=Firstnode;
    while(head!=NULL)
    {
        printf("%d->",head->data);
        head=head->next;
    }
    printf("NULL\n");
    return 0;
}
```

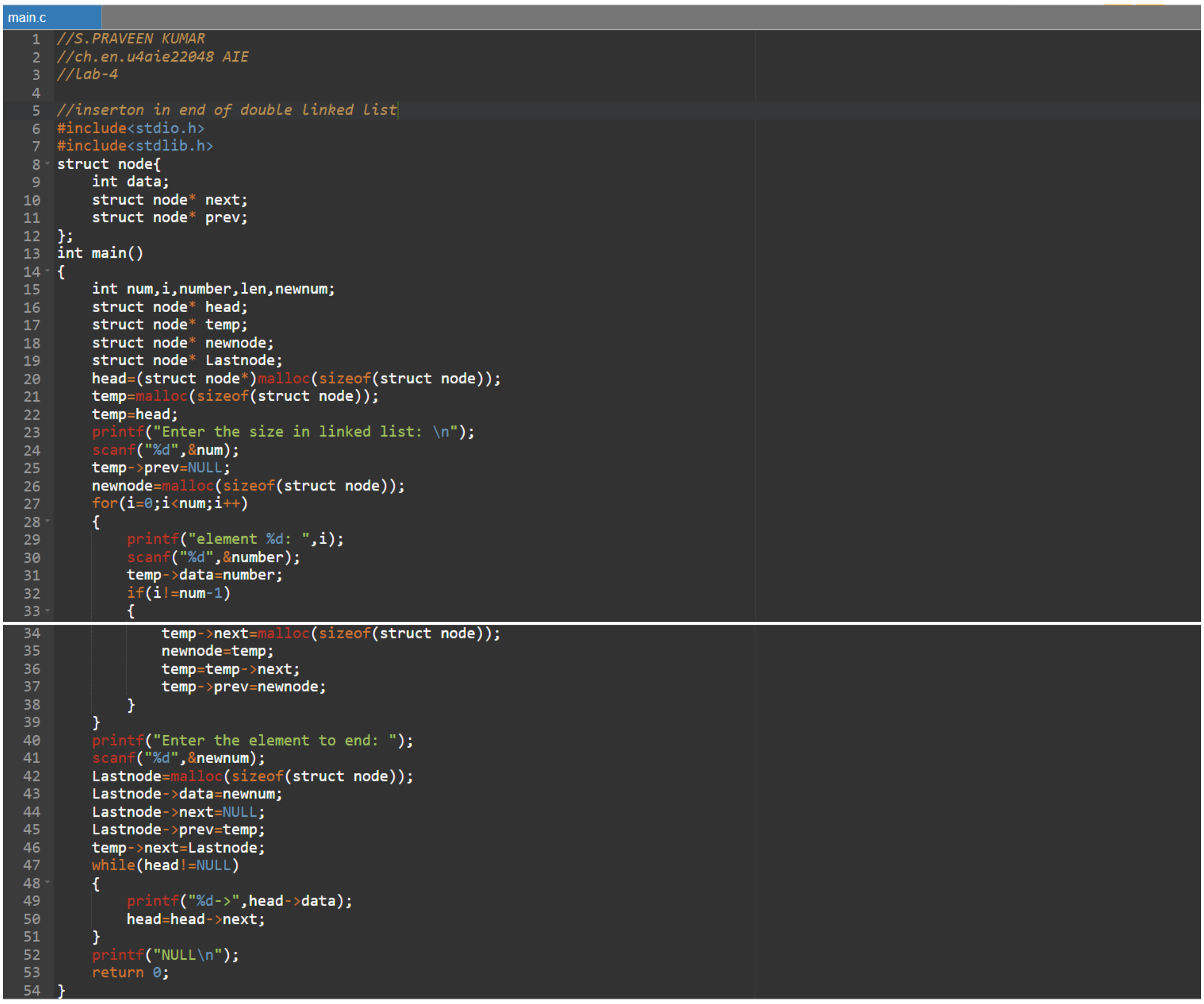
Output:



```
Enter the size in linked list:
5
element 0: 1
element 1: 2
element 2: 3
element 3: 4
element 4: 5
Enter the element to begin: 0
0->1->2->3->4->5->NULL


...Program finished with exit code 0
Press ENTER to exit console.
```

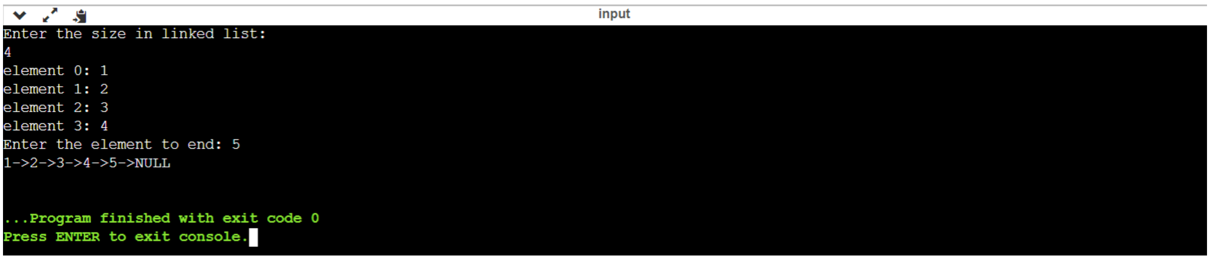
Insertion at end:

Program:



```c
//S.PRAVEEN KUMAR
//ch.en.u4aie22048 AIE
//lab-4

//inserton in end of double linked list
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node* next;
    struct node* prev;
};
int main()
{
    int num,i,number,len,newnum;
    struct node* head;
    struct node* temp;
    struct node* newnode;
    struct node* Lastnode;
    head=(struct node*)malloc(sizeof(struct node));
    temp=malloc(sizeof(struct node));
    temp=head;
    printf("Enter the size in linked list: \n");
    scanf("%d",&num);
    temp->prev=NULL;
    newnode=malloc(sizeof(struct node));
    for(i=0;i<num;i++)
    {
        printf("element %d: ",i);
        scanf("%d",&number);
        temp->data=number;
        if(i!=num-1)
        {
            temp->next=malloc(sizeof(struct node));
            newnode=temp;
            temp=temp->next;
            temp->prev=newnode;
        }
    }
    printf("Enter the element to end: ");
    scanf("%d",&newnum);
    Lastnode=malloc(sizeof(struct node));
    Lastnode->data=newnum;
    Lastnode->next=NULL;
    Lastnode->prev=temp;
    temp->next=Lastnode;
    while(head!=NULL)
    {
        printf("%d->",head->data);
        head=head->next;
    }
    printf("NULL\n");
    return 0;
}
```

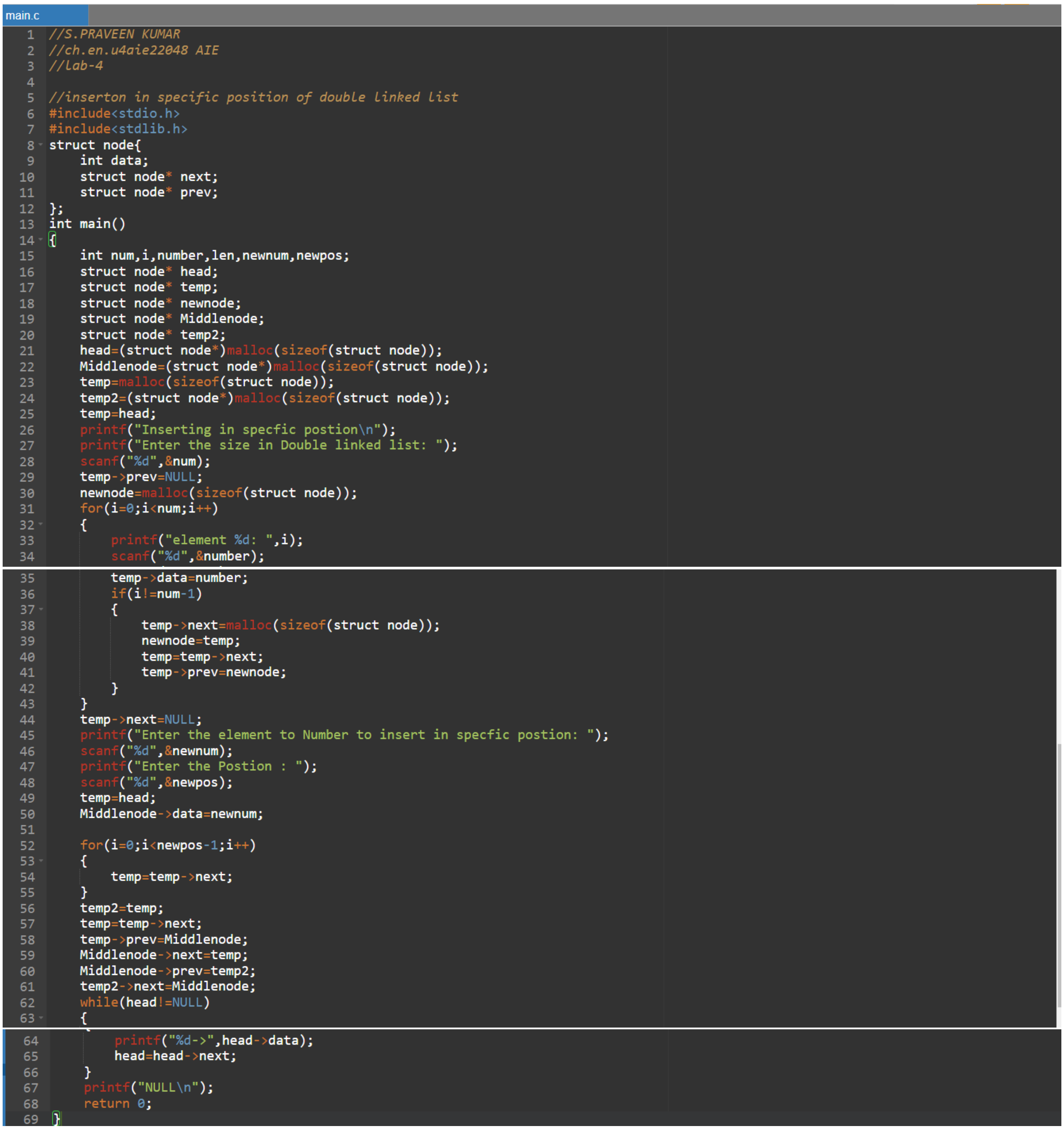
Output:



```
Enter the size in linked list:
4
element 0: 1
element 1: 2
element 2: 3
element 3: 4
Enter the element to end: 5
1->2->3->4->5->NULL


...Program finished with exit code 0
Press ENTER to exit console.
```

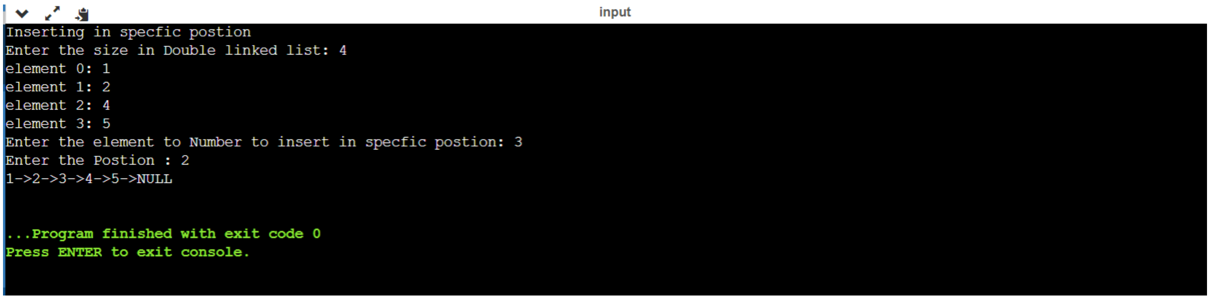
Inserting at specific position:

Program:



```c
//S.PRAVEEN KUMAR
//ch.en.u4aie22048 AIE
//lab-4

//inserton in specific position of double linked list
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node* next;
    struct node* prev;
};
int main()
{
    int num,i,number,len,newnum,newpos;
    struct node* head;
    struct node* temp;
    struct node* newnode;
    struct node* Middlenode;
    struct node* temp2;
    head=(struct node*)malloc(sizeof(struct node));
    Middlenode=(struct node*)malloc(sizeof(struct node));
    temp=malloc(sizeof(struct node));
    temp2=(struct node*)malloc(sizeof(struct node));
    temp=head;
    printf("Inserting in specfic postion\n");
    printf("Enter the size in Double linked list: ");
    scanf("%d",&num);
    temp->prev=NULL;
    newnode=malloc(sizeof(struct node));
    for(i=0;i<num;i++)
    {
        printf("element %d: ",i);
        scanf("%d",&number);
        temp->data=number;
        if(i!=num-1)
        {
            temp->next=malloc(sizeof(struct node));
            newnode=temp;
            temp=temp->next;
            temp->prev=newnode;
        }
    }
    temp->next=NULL;
    printf("Enter the element to Number to insert in specfic postion: ");
    scanf("%d",&newnum);
    printf("Enter the Postion : ");
    scanf("%d",&newpos);
    temp=head;
    Middlenode->data=newnum;

    for(i=0;i<newpos-1;i++)
    {
        temp=temp->next;
    }
    temp2=temp;
    temp=temp->next;
    temp->prev=Middlenode;
    Middlenode->next=temp;
    Middlenode->prev=temp2;
    temp2->next=Middlenode;
    while(head!=NULL)
    {
        printf("%d->",head->data);
        head=head->next;
    }
    printf("NULL\n");
    return 0;
}
```

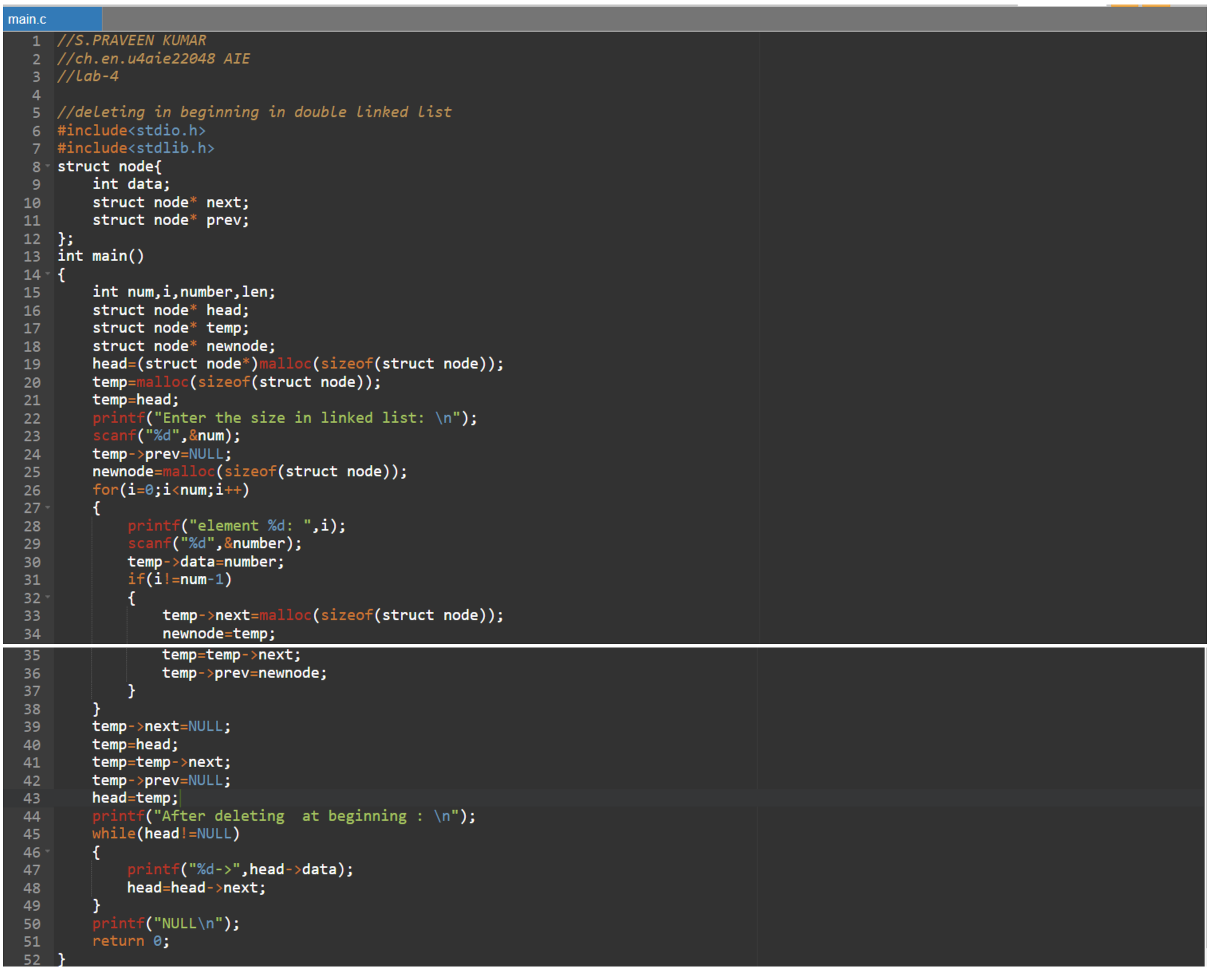
Output:



```
Inserting in specfic postion
Enter the size in Double linked list: 4
element 0: 1
element 1: 2
element 2: 4
element 3: 5
Enter the element to Number to insert in specfic postion: 3
Enter the Postion : 2
1->2->3->4->5->NULL


...Program finished with exit code 0
Press ENTER to exit console.
```

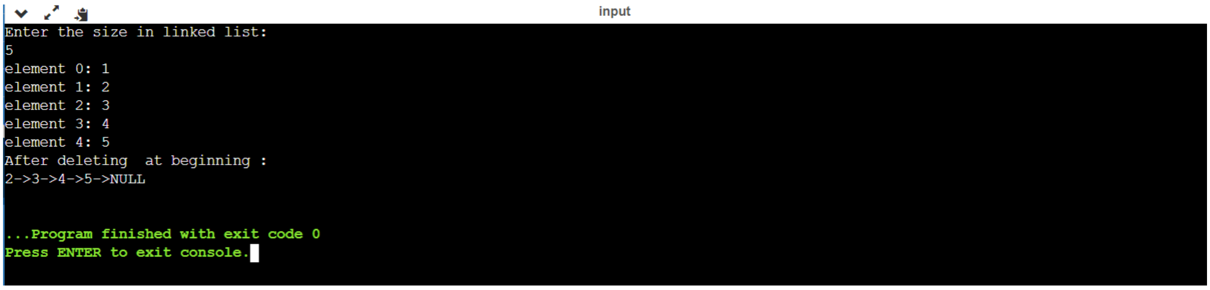
Deleting at Beginning:

Program:



```c
//S.PRAVEEN KUMAR
//ch.en.u4aie22048 AIE
//lab-4

//deleting in beginning in double linked list
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node* next;
    struct node* prev;
};
int main()
{
    int num,i,number,len;
    struct node* head;
    struct node* temp;
    struct node* newnode;
    head=(struct node*)malloc(sizeof(struct node));
    temp=malloc(sizeof(struct node));
    temp=head;
    printf("Enter the size in linked list: \n");
    scanf("%d",&num);
    temp->prev=NULL;
    newnode=malloc(sizeof(struct node));
    for(i=0;i<num;i++)
    {
        printf("element %d: ",i);
        scanf("%d",&number);
        temp->data=number;
        if(i!=num-1)
        {
            temp->next=malloc(sizeof(struct node));
            newnode=temp;
            temp=temp->next;
            temp->prev=newnode;
        }
    }
    temp->next=NULL;
    temp=head;
    temp=temp->next;
    temp->prev=NULL;
    head=temp;
    printf("After deleting  at beginning : \n");
    while(head!=NULL)
    {
        printf("%d->",head->data);
        head=head->next;
    }
    printf("NULL\n");
    return 0;
}
```

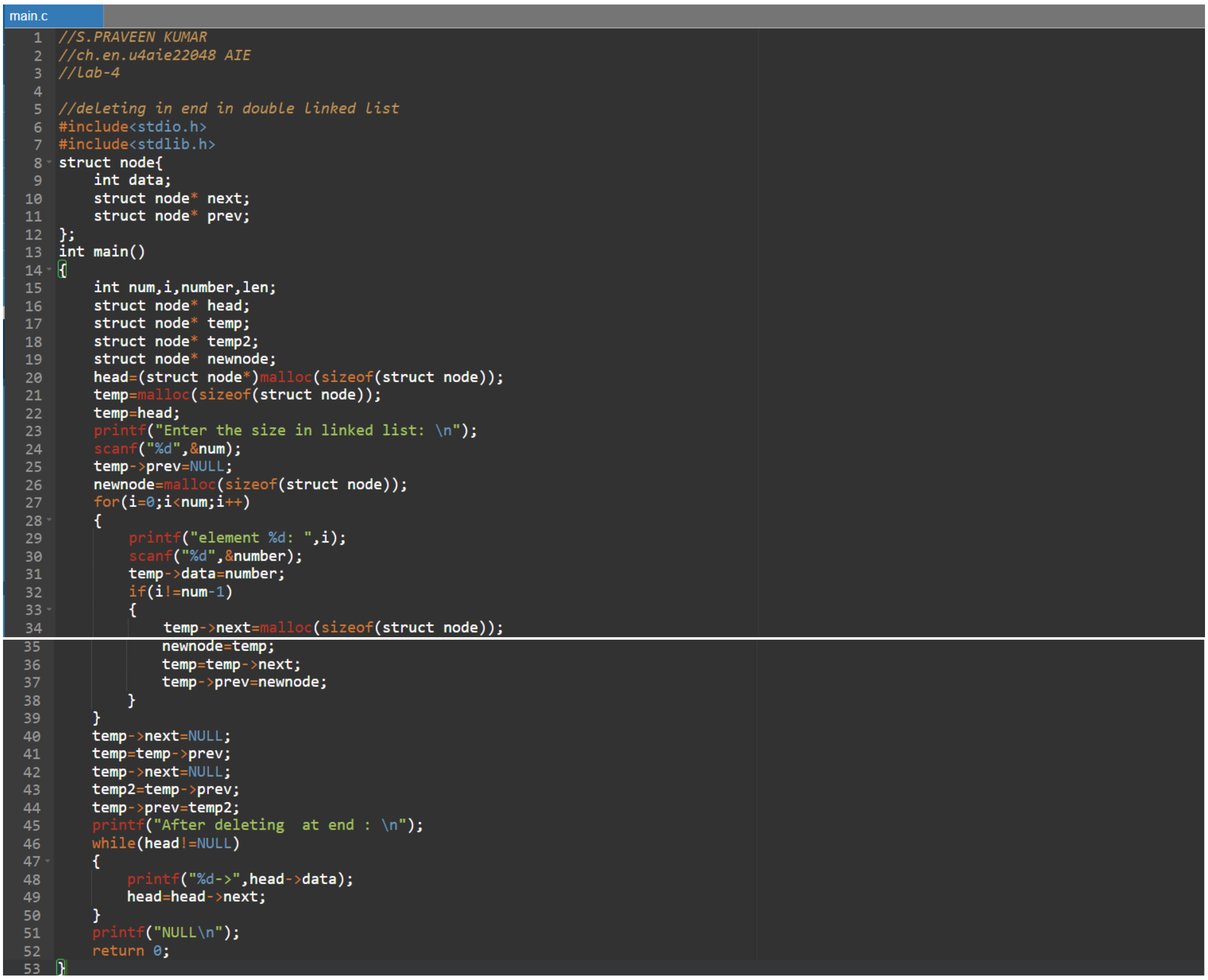
Output:



```
Enter the size in linked list:
5
element 0: 1
element 1: 2
element 2: 3
element 3: 4
element 4: 5
After deleting  at beginning :
2->3->4->5->NULL


...Program finished with exit code 0
Press ENTER to exit console.
```

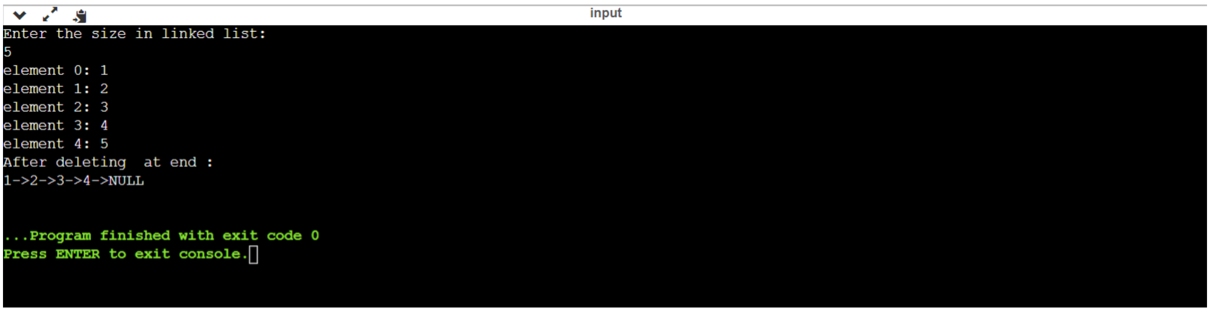
Deleting at end:

Program:



```c
//S.PRAVEEN KUMAR
//ch.en.u4aie22048 AIE
//lab-4

//deleting in end in double linked list
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node* next;
    struct node* prev;
};
int main()
{
    int num,i,number,len;
    struct node* head;
    struct node* temp;
    struct node* temp2;
    struct node* newnode;
    head=(struct node*)malloc(sizeof(struct node));
    temp=malloc(sizeof(struct node));
    temp=head;
    printf("Enter the size in linked list: \n");
    scanf("%d",&num);
    temp->prev=NULL;
    newnode=malloc(sizeof(struct node));
    for(i=0;i<num;i++)
    {
        printf("element %d: ",i);
        scanf("%d",&number);
        temp->data=number;
        if(i!=num-1)
        {
            temp->next=malloc(sizeof(struct node));
            newnode=temp;
            temp=temp->next;
            temp->prev=newnode;
        }
    }
    temp->next=NULL;
    temp=temp->prev;
    temp->next=NULL;
    temp2=temp->prev;
    temp->prev=temp2;
    printf("After deleting  at end : \n");
    while(head!=NULL)
    {
        printf("%d->",head->data);
        head=head->next;
    }
    printf("NULL\n");
    return 0;
}
```

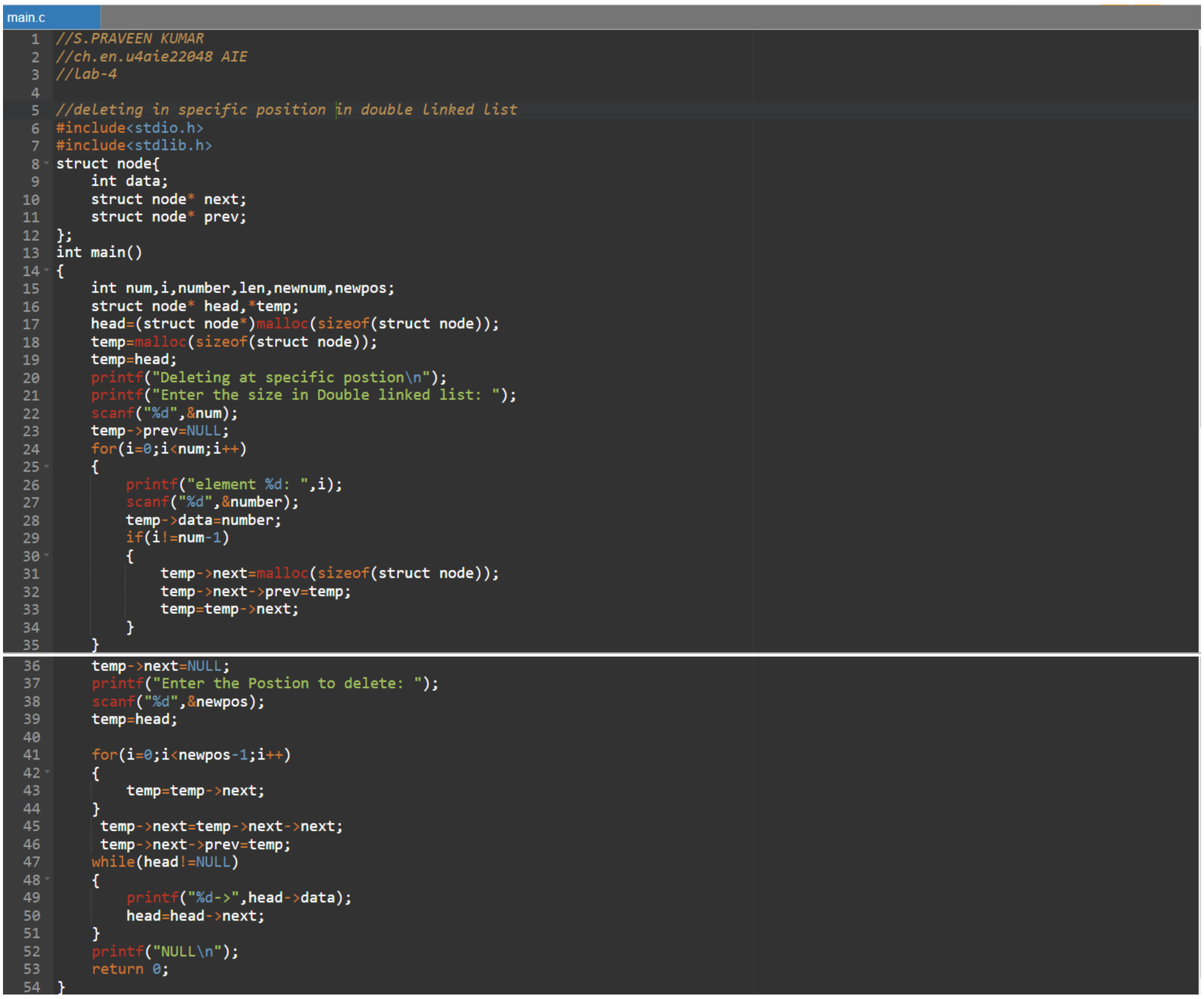
Output:



```
Enter the size in linked list:
5
element 0: 1
element 1: 2
element 2: 3
element 3: 4
element 4: 5
After deleting  at end :
1->2->3->4->NULL


...Program finished with exit code 0
Press ENTER to exit console.
```

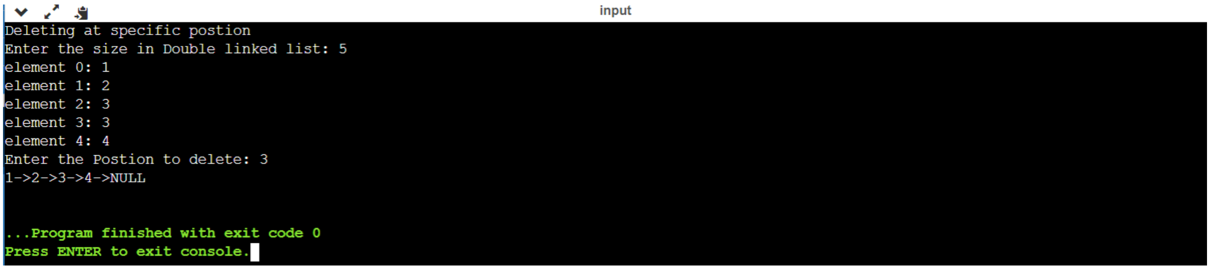
Deletion at specific position:

Program:



```c
//S.PRAVEEN KUMAR
//ch.en.u4aie22048 AIE
//lab-4

//deleting in specific position in double linked list
#include<stdio.h>
#include<stdlib.h>
struct node{
    int data;
    struct node* next;
    struct node* prev;
};
int main()
{
    int num,i,number,len,newnum,newpos;
    struct node* head,*temp;
    head=(struct node*)malloc(sizeof(struct node));
    temp=malloc(sizeof(struct node));
    temp=head;
    printf("Deleting at specific postion\n");
    printf("Enter the size in Double linked list: ");
    scanf("%d",&num);
    temp->prev=NULL;
    for(i=0;i<num;i++)
    {
        printf("element %d: ",i);
        scanf("%d",&number);
        temp->data=number;
        if(i!=num-1)
        {
            temp->next=malloc(sizeof(struct node));
            temp->next->prev=temp;
            temp=temp->next;
        }
    }
    temp->next=NULL;
    printf("Enter the Postion to delete: ");
    scanf("%d",&newpos);
    temp=head;

    for(i=0;i<newpos-1;i++)
    {
        temp=temp->next;
    }
     temp->next=temp->next->next;
     temp->next->prev=temp;
    while(head!=NULL)
    {
        printf("%d->",head->data);
        head=head->next;
    }
    printf("NULL\n");
    return 0;
}
```

Output:



```
Deleting at specific postion
Enter the size in Double linked list: 5
element 0: 1
element 1: 2
element 2: 3
element 3: 3
element 4: 4
Enter the Postion to delete: 3
1->2->3->4->NULL


...Program finished with exit code 0
Press ENTER to exit console.
```