

## DATA STRUCTURE – 1

### LAB-1

S.Praveen kumar  
ch.en.u4aie22048

### Traverse

#### Program:

```
1 //S.Praveen Kumar
2 //AIE ch.en.u4aie22048
3 //Data structure Lab-1
4
5 // Traverse
6
7 #include <stdio.h>
8 int main()
9 {
10     int i,n;
11     printf("enter size of the array: ");
12     scanf("%d",&n);
13     int array[n];
14     printf("enter the elements of array:\n");
15     for(i=0;i<n;i++)
16     {
17         printf("a[%d]: ",i);
18         scanf("%d",&array[i]);
19     }
20     printf("The array is-----:");
21     for(i=0;i<n;i++)
22     {
23         printf("%d\n",array[i]);
24     }
25
26     return 0;
27 }
```

#### 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:

```
1 //S.Praveen Kumar
2 //AIE ch.en.u4aie22048
3 //Data Structure Lab-1
4
5 //Insertion
6 #include<stdio.h>
7 int main()
8 {
9     int case1,array[100], n,i, item,pos, size, val,values;
10    printf("1.Insertation at Beginning\n2.Insertion at specific Position\n3.Insertion at end\n");
11    printf("Enter a Number: ");
12    scanf("%d",&case1);
13    switch(case1)
14    {
15        case 1:
16        {
17            printf("Enter the size of array: ");
18            scanf("%d", &n);
19            printf("\nEnter Elements in array: \n");
20            for(i=0;i<n;i++)
21            {
22                printf("a[%d]: ",i);
23                scanf("%d", &array[i]);
24            }
25            printf("enter the element at the beginning\n");
26            scanf("%d", &item);
27            n++;
28            for(i=n; i>1; i--)
29            {
30                array[i-1]=array[i-2];
31            }
32            array[0]=item;
33            printf("After resultant array element-----\n");
34            for(i=0;i<n;i++)
35            {
36                printf("\n%d", array[i]);
37            }
38            break;
39        }
40        case 2:
41        {
42            printf("Enter size of the array: ");
43            scanf("%d", &size);
44            printf("Enter elements: \n");
45            for (int i = 0; i < size; i++)
46            {
47                printf("a[%d]: ",i);
48                scanf("%d", &array[i]);
49            }
50            printf("Enter the insertion location\n");
51            scanf("%d", &pos);
52            printf("Enter the value to insert\n");
53            scanf("%d", &val);
54            for (int i = size - 1; i >= pos - 1; i--)
55            {
56                array[i+1] = array[i];
57            }
58            array[pos-1] = val;
59            printf("After inserting Resultant array is-----\n");
60            for (int i = 0; i <= size; i++)
61            {
62                printf("%d\n", array[i]);
63            }
64            break;
65        }
66        case 3:
67        {
68            printf("Enter size of Array Elements: ");
69            scanf("%d",&n);
70            int array[n];
71            printf("Enter the element of array:\n");
72            for(i=0; i<n; i++)
73            {
74                printf("a[%d]: ",i);
75                scanf("%d", &array[i]);
76            }
77            n++;
78            printf("\nEnter Element to Insert: ");
79            scanf("%d", &values);
80            array[i] = values;
81            printf("\nThe After adding the last element-----\n");
82            for(i=0; i<n; i++)
83            {
84                printf("%d\n", array[i]);
85            }
86            break;
87        }
88        default:
89        {
90            break;
91        }
92    }
93    return 0;
94 }
95
96 }
```

## Output:

```
input
1.Insertation 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.Insertation 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.

input
1.Insertation 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:

```
1 //S.Praveen Kumar
2 //AIE ch.en.u4aie22048
3 //Data structure lab-1
4
5 // Deletion
6
7 #include<stdio.h>
8
9 int main()
10 {
11     int case1;
12     printf("1.Deletion at beginning\n2.Deletion at specific point\n3.Deletion at end\n");
13     printf("Enter a number: ");
14     scanf("%d",&case1);
15     switch(case1)
16     {
17         case 1:
18             int n,array[10];
19             printf("enter the size of an array:");
20             scanf("%d",&n);
21             printf("enter elements in an array: \n");
22             for(int i=0;i<n;i++)
23             {
24                 printf("a[%d]:",i);
25                 scanf("%d",&array[i]);
26             }
27             n--;
28             for(int i=0;i<n;i++)
29             {
30                 array[i]=array[i+1];
31             }
32             printf("\nafter deletion----- \n");
33             for(int i=0;i<n;i++)
34             {
35                 printf("%d\n", array[i]);
36             }
37             break;
38
39         case 2:
40         {
41             int arr[10];
42             int pos,i,num;
43             printf("Enter the number of elements in an array: \n ");
44             scanf("%d",&num);
45             printf("Enter %d elements in array: \n ", num);
46             for (i = 0; i < num; i++)
47             {
48                 printf("arr[%d]: ", i);
49                 scanf("%d",&arr[i]);
50             }
51             printf("the position of the array element to delete: \n");
52             scanf("%d",&pos);
53             if (pos >= num+1)
54             {
55                 printf("Deletion is not possible in the array.\n");
56             }
57             else
58             {
59                 for (i = pos - 1; i < num -1; i++)
60                 {
61                     arr[i] = arr[i+1];
62                 }
63                 printf("The resultant array is-----: \n");
64                 for (i = 0; i < num - 1; i++)
65                 {
66                     printf("%d \n", arr[i]);
67                 }
68                 break;
69             }
70         }
71         case 3:
72             printf("enter the size of an array\n");
73             scanf("%d",&n);
74             printf("enter elements in an array\n");
75             for(int i=0;i<n;i++)
76             {
77                 printf("a[%d]: ",i);
78                 scanf("%d",&array[i]);
79             }
80             printf("\nafter deletion array elements are-----");
81             for(int i=0;i<n-1;i++)
82             {
83                 printf("\n%d", array[i]);
84             }
85             break;
86             default:
87             {
88                 break;
89             }
90     }
91     return 0;
92 }
```

## Output:

input

```
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.
```

input

```
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.
```

input

```
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

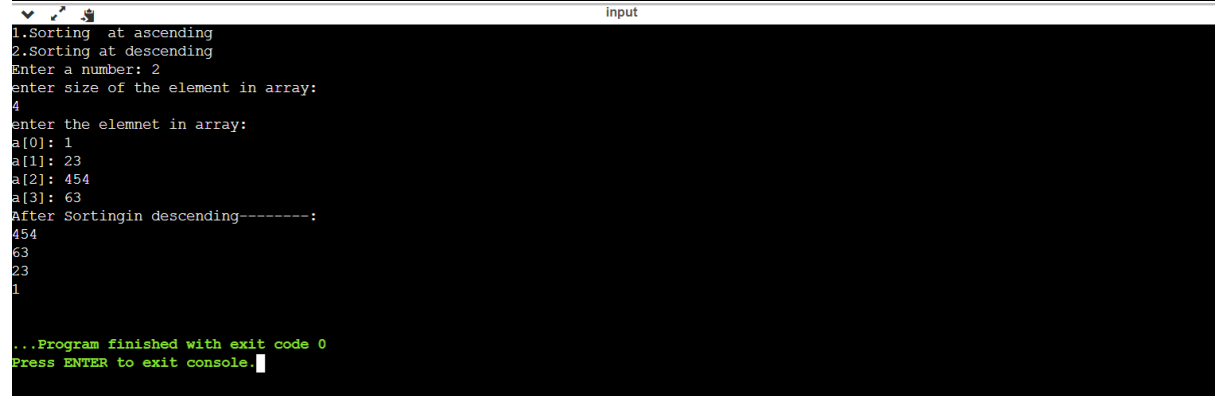
### Program:

```
main.c
1 //S.Praveen Kumar
2 //AIE ch.en.u4aie22048
3 //Data structure Lab-1
4
5 // Deletion
6
7 #include<stdio.h>
8
9 int main()
10 {
11     int case1;
12     printf("1.Sorting at ascending\n2.Sorting at descending\n");
13     printf("Enter a number: ");
14     scanf("%d",&case1);
15     switch(case1)
16     {
17         case 1:
18         {
19             int i,n,j,a;
20             printf("enter size of the element in array: \n");
21             scanf("%d",&n);
22             int array[n];
23             printf("enter the elemnet in array: \n");
24             for(i=0;i<n;i++)
25             {
26                 printf("a[%d]: ",i);
27                 scanf("%d",&array[i]);
28             }
29             for(i=0;i<n;i++)
30             {
31                 for(j=0;j<n;j++)
32                 {
33                     if(array[i]<array[j])
34                     {
35                         a=array[i];
36                         array[i]=array[j];
37                         array[j]=a;
38                     }
39                 }
40             }
41             printf("After Sortingin ascending-----:\n");
42             for(i=0;i<n;i++)
43             {
44                 printf("%d\n",array[i]);
45             }
46             break;
47         }
48         case 2:
49         {
50             int i,n,j,a;
51             printf("enter size of the element in array: \n");
52             scanf("%d",&n);
53             int array[n];
54             printf("enter the elemnet in array: \n");
55             for(i=0;i<n;i++)
56             {
57                 printf("a[%d]: ",i);
58                 scanf("%d",&array[i]);
59             }
60             for(i=0;i<n;i++)
61             {
62                 for(j=0;j<n;j++)
63                 {
64                     if(array[i]>array[j])
65                     {
66                         a=array[i];
67                         array[i]=array[j];
68                         array[j]=a;
69                     }
70                 }
71             }
72             printf("After Sortingin descending-----:\n");
73             for(i=0;i<n;i++)
74             {
75                 printf("%d\n",array[i]);
76             }
77             break;
78         }
79         default:
80         {
81             break;
82         }
83     }
84     return 0;
85 }
```

### 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.
```



The screenshot shows a terminal window with a title bar containing icons for window control and the word "input". The terminal displays the following text:

```
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:

```
1 //S.Praveen Kumar
2 //AIE ch.en.u4aie22048
3 //Data Structures Lab-1
4
5 //searching
6
7 #include <stdio.h>
8 int main()
9 {
10     int i,n,element,count=0;
11     printf("Enter size of the array : ");
12     scanf("%d", &n);
13     int a[n];
14     printf("Enter elements in array : \n");
15     for(i=0; i<n; i++)
16     {
17         printf("a[%d]: ",i);
18         scanf("%d",&a[i]);
19     }
20     printf("Enter the element to search: ");
21     scanf("%d", &element);
22
23     for(i=0; i<n; i++)
24     {
25         if(a[i]==element)
26         {
27             printf("element found ");
28             count=count+1;
29         }
30     }
31
32     if(count==0)
33     {
34         printf("Elemnet not found");
35     }
36     return 0;
37 }
```

### Output:

```
Input
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
1 //S.Praveen kumar
2 //AIE ch.en.u4aie22048
3 // Lab-2
4
5 //declare and initialize
6
7 #include <stdio.h>
8 int main()
9 {
10     int m,n,i,j;
11     printf("Enter the row size of the array: ");
12     scanf("%d",&m);
13     printf("Enter the column size of the array: ");
14     scanf("%d",&n);
15     int a[m][n]; //declare
16     for(i=0;i<m;i++)
17     {
18         for(j=0;j<n;j++)
19         {
20             printf("Enter the element a[%d][%d]: ",i,j); //initialing
21             scanf("%d",&a[i][j]);
22         }
23     }
24     printf("The 2d array is -----\\n");
25     for(i=0;i<m;i++)
26     {
27         for(j=0;j<n;j++)
28         {
29             printf(" %d ",a[i][j]);
30             printf("\\n");
31         }
32     }
33
34     return 0;
35 }
36
```

#### 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

```
main.c
1 //S.Praveen Kumar
2 //AIE ch.en.u4aie22048
3 //Data structures Lab 2
4
5 //sum of 3x3 matrix
6 #include<stdio.h>
7 int main()
8 {
9     int i,j,row,col,e,ele,posi,posj;
10    printf("Enter row size: ");
11    scanf("%d",&row);
12    printf("Enter column size: ");
13    scanf("%d",&col);
14    int a[row][col];
15    printf("Enter the elements of array: \n");
16    for(i=0; i<row; i++)
17    {
18        for(j=0; j<col; j++)
19        {
20            printf("a[%d][%d]: ",i,j);
21            scanf("%d",&a[i][j]);
22        }
23    }
24    printf("\nEnter the element: ");
25    scanf("%d",&ele);
26    printf("\nEnter the row positions : ");
27    scanf("%d",&posi);
28    printf("\nEnter the col positions : ");
29    scanf("%d",&posj);
30    for(i=row-1;i>=posi;i--)
31    {
32        for(j=col-1;j>=posj;j--)
33        {
34            a[i+1][j+1] = a[i][j];
35            a[posi][posj] = ele;
36        }
37    }
38    for(i=0; i<row; i++)
39    {
40        for(j=0; j<col; j++)
41        {
42            printf("%d",a[i][j]);
43        }
44    }
45
46    return 0;
47 }
48
49
50
51
```

### Output

```
input
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
1 //S.Praveen kumar
2 //AIE ch.en.udate22048
3 // Lab-2
4
5 //updating a 2d array
6
7 #include <stdio.h>
8 int main()
9 {
10     int m,n,i,j,num,nm,nn;
11     printf("Enter the row size of the array: ");
12     scanf("%d",&m);
13     printf("Enter the column size of the array: ");
14     scanf("%d",&n);
15     int a[m][n],a1[m][n];
16     for(i=0;i<m;i++)
17     {
18         for(j=0;j<n;j++)
19         {
20             printf("Enter the element a[%d][%d]: ",i,j);
21             scanf("%d",&a[i][j]);
22         }
23     }
24     for(i=0; i<m ;i++)
25     {
26         for(j=0; j<n ;j++)
27         {
28             a1[i][j]=a[i][j];
29         }
30     }
31     printf("Enter the element to update: ");
32     scanf("%d",&num);
33     printf("Enter the position of element to update: ");
34     scanf("%d %d",&nm,&nn);
35     for(i=0; i<m ;i++)
36     {
37         for(j=0; j<n ;j++)
38         {
39             if(i==nm-1 && j==nn-1)
40             {
41                 a1[i][j]=num;
42             }
43         }
44     }
45     printf("After the updating-----\n");
46     for(i=0; i<m ;i++)
47     {
48         for(j=0; j<n ;j++)
49         {
50             printf("%d ",a1[i][j]);
51         }
52         printf("\n");
53     }
54     return 0;
55 }
```

#### 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:

```
main.c
1 //S.Praveen kumar
2 //AIE ch.en.u4aie22048
3 // Lab-2
4
5 //delete a row
6
7 #include <stdio.h>
8 int main()
9 {
10     int m,n,i,j,num;
11     printf("Enter the row size of the array: ");
12     scanf("%d",&m);
13     printf("Enter the column size of the array: ");
14     scanf("%d",&n);
15     int a[m][n],a1[m][n];
16     for(i=0;i<m;i++)
17     {
18         for(j=0;j<n;j++)
19         {
20             printf("Enter the element a[%d][%d]: ",i,j);
21             scanf("%d",&a[i][j]);
22         }
23     }
24     for(i=0;i<m;i++)
25     {
26         for(j=0;j<n;j++)
27         {
28             a1[i][j]=a[i][j];
29         }
30     }
31     printf("Enter the row to delete: ");
32     scanf("%d",&num);
33     m--;
34     for(i=0;i<m;i++)
35     {
36         for(j=0;j<n;j++)
37         {
38             if(i==num-1)
39                 a1[i+1][j]=a[i][j];
40         }
41     }
42     printf("After the deltion -----\\n");
43     for(i=0;i<m;i++)
44     {
45         for(j=0;j<n;j++)
46         {
47             printf(" %d ",a1[i][j]);
48         }
49         printf("\\n");
50     }
51     return 0;
52 }
53
54
```

### 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

```
main.c
1 //S.Praveen Kumar
2 //AIE ch.en.u4aie22048
3 //Data structures Lab 2
4
5 //sum of 3x3 matrix
6 #include <stdio.h>
7 int main()
8 {
9     int row,col,array1,row1,col1,arr1,arr2,i,j,k;
10    printf("Enter the size of row: ");
11    scanf("%d",&row);
12    printf("Enter the size of col: ");
13    scanf("%d",&col);
14    printf("Enter the size of array1: ");
15    scanf("%d",&arr1);
16    printf("Enter the size of row1: ");
17    scanf("%d",&row1);
18    printf("Enter the size of col: ");
19    scanf("%d",&col1);
20    printf("Enter the size of array2: ");
21    scanf("%d",&arr2);
22    int a[row][col][arr1],b[row1][col1][arr2],sum[row][col1][arr2];
23    printf("Enter the elements of the array: ");
24    for(i=0;i<row;i++)
25    {
26        for(j=0;j<col;j++)
27        {
28            for(k=0;k<arr1;k++)
29            {
30                printf("a[%d][%d][%d]: ",i,j,k);
31                scanf("%d",&a[i][j][k]);
32            }
33        }
34    }
35
36    printf("enter the 2nd array values: \n");
37    for(i=0;i<row1;i++)
38    {
39        for(j=0;j<col1;j++)
40        {
41            for(k=0;k<arr1;k++)
42            {
43                printf("b[%d][%d][%d]: ",i,j,k);
44                scanf("%d",&b[i][j][k]);
45            }
46        }
47    }
48
49    for(i=0;i<row;i++)
50    {
51        for(j=0;j<col1;j++)
52        {
53            for(k=0;k<arr2;k++)
54            {
55                sum[i][j][k]=a[i][j][k]+b[i][j][k];
56            }
57        }
58    }
59
60    printf("Sum of 3x3 array is-----: \n");
61    for(i=0;i<row1;i++)
62    {
63        for(j=0;j<col1;j++)
64        {
65            for(k=0;k<arr1;k++)
66            {
67                printf("%d ",sum[i][j][k]);
68            }
69            printf("\n");
70        }
71    }
72
73    return 0;
74 }
75
76
```

### Output:

```
input
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:**

```
main.c
1 //S.PRAVEEN KUMAR
2 //AIE ch.en.u4aie22048
3 //Lab -2
4
5 //SORTING row wise and column wise
6 #include <stdio.h>
7
8 void main () {
9
10     static int ma[10][10],mb[10][10];
11
12     int i,j,k,a,m,n;
13
14     printf ("Enter the size of the matrix \n");
15
16     scanf ("%d %d", &m,&n);
17
18     printf ("Enter element of the matrix \n");
19
20     for (i=0;i<m;i++)
21     {
22         for (j=0;j<n;j++)
23         {
24             scanf ("%d",&ma[i][j]);
25             mb[i][j] = ma[i][j];
26         }
27     }
28
29     printf ("The given matrix is \n");
30
31     for (i=0;i<m;i++)
32     {
33         for (j=0;j<n;j++)
34         {
35             printf (" %d",ma[i][j]);
36         }
37         printf ("\n");
38     }
39
40     printf ("After arranging rows in ascending order\n");
41
42     for (i=0;i<m;i++) {
43         for (j=0;j<n;j++) {
44             for (k=j+1;k<n;k++) {
45                 if (ma[i][j] > ma[i][k]) {
46                     a = ma[i][j];
47                     ma[i][j] = ma[i][k];
48                     ma[i][k] = a;
49                 }
50             }
51         }
52     }
53
54     for (i=0;i<m;i++)
55     {
56         for (j=0;j<n;j++)
57         {
58             printf (" %d",ma[i][j]);
59         }
60         printf ("\n");
61     }
62     printf ("After arranging the columns in descending order \n");
63     for (j=0;j<n;j++)
64     {
65         for (i=0;i<m;i++)
66         {
67             for (k=i+1;k<m;k++)
68             {
69                 if (mb[i][j] < mb[k][j])
70                 {
71                     a = mb[i][j];
72                     mb[i][j] = mb[k][j];
73                     mb[k][j] = a;
74                 }
75             }
76         }
77     }
78     for (i=0;i<m;i++) {
79         for (j=0;j<n;j++) {
80
81             printf (" %d",mb[i][j]);
82
83         }
84
85         printf ("\n");
86     }
87
88 }
89
90 }
```

### OUTPUT:

```
input
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:

```
main.c
1 //S.Praveen kumar
2 //AIE ch.en.u4aie22048
3 //data structures lab-3
4
5 // initialize and declaration of Link list;
6
7 #include<stdio.h>
8 #include<stdlib.h>
9 struct node
10 {
11     int data;
12     struct node* next;
13 };
14
15 int main()
16 {
17     struct node* head;
18     struct node* temp;
19     head=malloc(sizeof(struct node));
20     temp=malloc(sizeof(struct node));
21     temp=head;
22     int num,number,i;
23     printf("Creating a link list: \n");
24     printf("Enter the size of linked list: ");
25     scanf("%d",&num);
26     for(i=0;i<num;i++)
27     {
28         printf("enter the element %d:",i);
29         scanf("%d",&number);
30         temp->data=number;
31         if(i<num-1)
32         {
33             temp->next=malloc(sizeof(struct node));
34             temp=temp->next;
35         }
36     }
37     temp->next=NULL;
38     printf("The LINK LIST -----> \n");
39     while(head!=NULL)
40     {
41         printf("%d->",head->data);
42         head=head->next;
43     }
44     return 0;
45 }
46
47
```

#### Output:

```
input
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:

```
main.c
1  #include<stdio.h>
2  #include<stdlib.h>
3  struct node{
4      int data;
5      struct node* next;
6      struct node* prev;
7  };
8  int main()
9  {
10     int num,i,number,len,newnum;
11     struct node* head;
12     struct node* temp;
13     struct node* newnode;
14     struct node* Firstnode;
15     head=(struct node*)malloc(sizeof(struct node));
16     temp=malloc(sizeof(struct node));
17     temp=head;
18     printf("Enter the size in linked list: \n");
19     scanf("%d",&num);
20     temp->prev=NULL;
21     newnode=malloc(sizeof(struct node));
22     for(i=0;i<num;i++)
23     {
24         printf("element %d: ",i);
25         scanf("%d",&number);
26         temp->data=number;
27         if(i!=num-1)
28         {
29             temp->next=malloc(sizeof(struct node));
30             newnode=temp;
31             temp=temp->next;
32             temp->prev=newnode;
33         }
34     }
35     temp->next=NULL;
36     printf("Enter the element to begin: ");
37     scanf("%d",&newnum);
38     Firstnode=malloc(sizeof(struct node));
39     Firstnode->data=newnum;
40     Firstnode->prev=NULL;
41     Firstnode->next=head;
42     head=Firstnode;
43     while(head!=NULL)
44     {
45         printf("%d->",head->data);
46         head=head->next;
47     }
48     printf("\n\n");
49     return 0;
50 }
```

### Output:

```
input
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:

```
main.c
1 #include<stdio.h>
2 #include<stdlib.h>
3 struct node{
4     int data;
5     struct node* next;
6     struct node* prev;
7 };
8 int main()
9 {
10     int num,i,number,len,newnum;
11     struct node* head;
12     struct node* temp;
13     struct node* newnode;
14     struct node* Lastnode;
15     head=(struct node*)malloc(sizeof(struct node));
16     temp=malloc(sizeof(struct node));
17     temp=head;
18     printf("Enter the size in linked list: \n");
19     scanf("%d",&num);
20     temp->prev=NULL;
21     newnode=malloc(sizeof(struct node));
22     for(i=0;i<num;i++)
23     {
24         printf("element %d: ",i);
25         scanf("%d",&number);
26         temp->data=number;
27         if(i!=num-1)
28         {
29             temp->next=malloc(sizeof(struct node));
30             newnode=temp;
31             temp=temp->next;
32             temp->prev=newnode;
33         }
34     }
35     printf("Enter the element to end: ");
36     scanf("%d",&newnum);
37     Lastnode=malloc(sizeof(struct node));
38     Lastnode->data=newnum;
39     Lastnode->next=NULL;
40     Lastnode->prev=temp;
41     temp->next=Lastnode;
42     while(head!=NULL)
43     {
44         printf("%d->",head->data);
45         head=head->next;
46     }
47     printf("NULL\n");
48     return 0;
49 }
```

## Output:

```
input
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:

```
main.c
1  #include<stdio.h>
2  #include<stdlib.h>
3  struct node{
4      int data;
5      struct node* next;
6  };
7  int main()
8  {
9      int num,i,number,len,number1,pos;
10     struct node* head;
11     struct node* temp;
12     struct node* temp1;
13     struct node* temp2;
14     head=(struct node*)malloc(sizeof(struct node));
15     temp=malloc(sizeof(struct node));
16     temp=head;
17     printf("Enter the size in linked list: \n");
18     scanf("%d",&num);
19     for(i=0;i<num;i++)
20     {
21         printf("element %d: ",i);
22         scanf("%d",&number);
23         temp->data=number;
24         if(i!=num-1)
25         {
26             temp->next=malloc(sizeof(struct node));
27             temp=temp->next;
28         }
29     }
30     temp->next=NULL;
31     temp1=malloc(sizeof(struct node));
32     printf("Enter the elemenet to add: ");
33     scanf("%d",&number1);
34     printf("Enter the position to add: ");
35     scanf("%d",&pos);
36     temp1->data=number1;
37     temp=head;
38     for(i=0;i<pos-1;i++)
39     {
40         temp=temp->next;
41     }
42     temp2=temp;
43     temp=temp->next;
44     temp2->next=temp1;
45     temp1->next=temp;
46
47     while(head!=NULL)
48     {
49         printf("%d->",head->data);
50         head=head->next;
51     }
52     printf("NULL");
53     return 0;
54 }
```

### Output:

```
input
Enter the size in linked list:
4
element 0: 1
element 1: 2
element 2: 4
element 3: 5
Enter the elemenet 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:

```
main.c
1 //S.Praveen Kumar
2 //AIE ch.u4.aie22048
3 //Lab-3
4
5 //Deletion at begin
6 #include<stdio.h>
7 #include<stdlib.h>
8 int main()
9 {
10     struct node{
11         int data;
12         struct node *link;
13     };
14     struct node *temp,*head;
15     head=(struct node *)malloc(sizeof(struct node));
16     temp=head;
17     int n,number,i;
18     printf("Enter the size of the linked list: ");
19     scanf("%d",&n);
20     for(i=0;i<n;i++)
21     {
22         printf("Enter the element %d: ",i);
23         scanf("%d",&number);
24         temp->data=number;
25         if(i!=n-1)
26         {
27             temp->link=(struct node*)malloc(sizeof(struct node));
28             temp=temp->link;
29         }
30     }
31     temp->link=NULL;
32     struct node *forward,*print;
33     forward=head;
34     forward->forward->link;
35     while(forward!=NULL)
36     {
37         printf("%d->",forward->data);
38         forward=forward->link;
39     }
40     printf("NULL\n");
41     return 0;
42 }
```

### Output:

```
Enter the size of the linked list: 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.
```

input

## Deletion at end:

### Program:

```
main.c
1 //S.Praveen Kumar
2 //AIE ch.u4.aie22048
3 //Lab-3
4
5 //Deletion at end
6 #include<stdio.h>
7 #include<stdlib.h>
8 int main()
9 {
10     struct node
11     {
12         int data;
13         struct node *link;
14     };
15     struct node *head,*temp;
16     head=(struct node *)malloc(sizeof(struct node));
17     temp=head;
18     int i,n,number;
19     printf("Enter the size of the linked list: ");
20     scanf("%d",&n);
21     for(i=0;i<n;i++)
22     {
23         printf("Enter the element %d: ",i);
24         scanf("%d",&number);
25         temp->data=number;
26         if(i!=n-1)
27         {
28             temp->link=(struct node *)malloc(sizeof(struct node));
29             temp=temp->link;
30         }
31     }
32     temp->link=NULL;
33     struct node *forward;
34     forward=head;
35     for(i=0;i<(n-1);i++)
36     {
37         printf("%d->",forward->data);
38         forward=forward->link;
39     }
40     printf("NULL\n");
41     return 0;
42 }
```

### Output:

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

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

## Deletion at specific position:

### Program:

```
main.c
1 //S.Praveen Kumar
2 //AIE ch.u4.aie22048
3 //Lab-3
4
5 //Deletion at specific position
6 #include<stdio.h>
7 #include<stdlib.h>
8 int main()
9 {
10     struct node
11     {
12         int data;
13         struct node *link;
14     };
15     struct node *head,*temp;
16     head=(struct node *)malloc(sizeof(struct node));
17     int n,number,i;
18     printf("Enter the size of the Linkedlist: ");
19     scanf("%d",&n);
20     temp=head;
21     for(i=0;i<n;i++)
22     {
23         printf("Enter the element %d: ",i);
24         scanf("%d",&number);
25         temp->data=number;
26         if(i!=n-1)
27         {
28             temp->link=(struct node *)malloc(sizeof(struct node));
29             temp=temp->link;
30         }
31     }
32     temp->link=NULL;
33     int pos;
34     printf("Enter the Postion: ");
35     scanf("%d",&pos);
36
37     struct node *transverse;
38     struct node *forward;
39     transverse=head;
40     forward=head;
41     for(i=0;i<(pos-1);i++)
42     {
43         forward=forward->link;
44     }
45     for(i=0;i<pos;i++)
46     {
47         transverse=transverse->link;
48     }
49     forward->link=transverse->link;
50     while(head!=NULL)
51     {
52         printf("%d",head->data);
53         head=head->link;
54     }
55     return 0;
56 }
```

### Output:

```
input
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:

```
main.c
1 //S.PRAVEEN KUMAR
2 //ch.en.u4aie22048 AIE
3 //Lab-4
4
5 //creation of double Linked List
6 #include<stdio.h>
7 #include<stdlib.h>
8 struct node{
9     int data;
10    struct node* next;
11    struct node* prev;
12 };
13 int main()
14 {
15     int num,i,number,len;
16     struct node* head;
17     struct node* temp;
18     head=(struct node*)malloc(sizeof(struct node));
19     temp=malloc(sizeof(struct node));
20     temp=head;
21     printf("Creation of double linked list\n Enter the size in linked list: \n");
22     scanf("%d",&num);
23     temp->prev=NULL;
24     for(i=0;i<num;i++)
25     {
26         printf("element %d: ",i);
27         scanf("%d",&number);
28         temp->data=number;
29         if(i!=num-1)
30         {
31             temp->next=malloc(sizeof(struct node));
32             temp->next->prev=temp;
33             temp=temp->next;
34         }
35     }
36     temp->next=NULL;
37     while(head!=NULL)
38     {
39         printf("%d->",head->data);
40         head=head->next;
41     }
42     printf("NULL\n");
43     return 0;
44 }
```

### Output:

```
input
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:

```
1 //S.PRAVEEN KUMAR
2 //ch.en.u4aie22048 AIE
3 //Lab-4
4
5 //insertion in beginning of double Linked List
6 #include<stdio.h>
7 #include<stdlib.h>
8 struct node{
9     int data;
10    struct node* next;
11    struct node* prev;
12 };
13 int main()
14 {
15     int num,i,number,len,newnum;
16     struct node* head;
17     struct node* temp;
18     struct node* newnode;
19     struct node* Firstnode;
20     head=(struct node*)malloc(sizeof(struct node));
21     temp=malloc(sizeof(struct node));
22     temp=head;
23     printf("Enter the size in linked list: \n");
24     scanf("%d",&num);
25     temp->prev=NULL;
26     newnode=malloc(sizeof(struct node));
27     for(i=0;i<num;i++)
28     {
29         printf("element %d: ",i);
30         scanf("%d",&number);
31         temp->data=number;
32
33         if(i!=num-1)
34         {
35             temp->next=malloc(sizeof(struct node));
36             newnode=temp;
37             temp=temp->next;
38             temp->prev=newnode;
39         }
40         temp->next=NULL;
41         printf("Enter the element to begin: ");
42         scanf("%d",&newnum);
43         Firstnode=malloc(sizeof(struct node));
44         Firstnode->data=newnum;
45         Firstnode->prev=NULL;
46         head->prev=Firstnode;
47         Firstnode->next=head;
48         head=Firstnode;
49         while(head!=NULL)
50         {
51             printf("%d->",head->data);
52             head=head->next;
53         }
54         printf("NULL\n");
55
56         printf("NULL\n");
57         return 0;
58     }
```

### Output:

```
input
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:

```
main.c
1 //S.PRAVEEN KUMAR
2 //ch.en.u4aie22048 AIE
3 //Lab-4
4
5 //insertion in end of double Linked List
6 #include<stdio.h>
7 #include<stdlib.h>
8 struct node{
9     int data;
10    struct node* next;
11    struct node* prev;
12 };
13 int main()
14 {
15     int num,i,number,len,newnum;
16     struct node* head;
17     struct node* temp;
18     struct node* newnode;
19     struct node* Lastnode;
20     head=(struct node*)malloc(sizeof(struct node));
21     temp=malloc(sizeof(struct node));
22     temp=head;
23     printf("Enter the size in linked list: \n");
24     scanf("%d",&num);
25     temp->prev=NULL;
26     newnode=malloc(sizeof(struct node));
27     for(i=0;i<num;i++)
28     {
29         printf("element %d: ",i);
30         scanf("%d",&number);
31         temp->data=number;
32         if(i!=num-1)
33         {
34             temp->next=malloc(sizeof(struct node));
35             newnode=temp;
36             temp=temp->next;
37             temp->prev=newnode;
38         }
39     }
40     printf("Enter the element to end: ");
41     scanf("%d",&newnum);
42     Lastnode=malloc(sizeof(struct node));
43     Lastnode->data=newnum;
44     Lastnode->next=NULL;
45     Lastnode->prev=temp;
46     temp->next=Lastnode;
47     while(head!=NULL)
48     {
49         printf("%d->",head->data);
50         head=head->next;
51     }
52     printf("NULL\n");
53     return 0;
54 }
```

### Output:

```
input
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:

```
main.c
1 //S.PRAVEEN KUMAR
2 //ch.en.u4aie22048 AIE
3 //Lab-4
4
5 //insertion in specific position of double linked list
6 #include<stdio.h>
7 #include<stdlib.h>
8 struct node{
9     int data;
10    struct node* next;
11    struct node* prev;
12 };
13 int main()
14 {
15     int num,i,number,len,newnum,newpos;
16     struct node* head;
17     struct node* temp;
18     struct node* newnode;
19     struct node* Middlednode;
20     struct node* temp2;
21     head=(struct node*)malloc(sizeof(struct node));
22     Middlednode=(struct node*)malloc(sizeof(struct node));
23     temp=malloc(sizeof(struct node));
24     temp2=(struct node*)malloc(sizeof(struct node));
25     temp=head;
26     printf("Inserting in specific position\n");
27     printf("Enter the size in Double linked list: ");
28     scanf("%d",&num);
29     temp->prev=NULL;
30     newnode=malloc(sizeof(struct node));
31     for(i=0;i<num;i++)
32     {
33         printf("element %d: ",i);
34         scanf("%d",&number);
35         temp->data=number;
36         if(i!=num-1)
37         {
38             temp->next=malloc(sizeof(struct node));
39             newnode=temp;
40             temp=temp->next;
41             temp->prev=newnode;
42         }
43     }
44     temp->next=NULL;
45     printf("Enter the element to Number to insert in specific position: ");
46     scanf("%d",&newnum);
47     printf("Enter the Position : ");
48     scanf("%d",&newpos);
49     temp=head;
50     Middlednode->data=newnum;
51     for(i=0;i<newpos-1;i++)
52     {
53         temp=temp->next;
54     }
55     temp2=temp;
56     temp=temp->next;
57     temp->prev=Middlednode;
58     Middlednode->next=temp;
59     Middlednode->prev=temp2;
60     temp2->next=Middlednode;
61     while(head!=NULL)
62     {
63         printf("%d->",head->data);
64         head=head->next;
65     }
66     printf("NULL\n");
67     return 0;
68 }
69 }
```

### Output:

```
Inserting in specific position
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 specific position: 3
Enter the Position : 2
1->2->3->4->5->NULL

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

## Deleting at Beginning:

### Program:

```
main.c
1 //S.PRAVEEN KUMAR
2 //ch.en.u4aie22048 AIE
3 //Lab-4
4
5 //deleting in beginning in double Linked List
6 #include<stdio.h>
7 #include<stdlib.h>
8 struct node{
9     int data;
10    struct node* next;
11    struct node* prev;
12 };
13 int main()
14 {
15     int num,i,number,len;
16     struct node* head;
17     struct node* temp;
18     struct node* newnode;
19     head=(struct node*)malloc(sizeof(struct node));
20     temp=malloc(sizeof(struct node));
21     temp=head;
22     printf("Enter the size in linked list: \n");
23     scanf("%d",&num);
24     temp->prev=NULL;
25     newnode=malloc(sizeof(struct node));
26     for(i=0;i<num;i++)
27     {
28         printf("element %d: ",i);
29         scanf("%d",&number);
30         temp->data=number;
31         if(i!=num-1)
32         {
33             temp->next=malloc(sizeof(struct node));
34             newnode=temp;
35             temp=temp->next;
36             temp->prev=newnode;
37         }
38     }
39     temp->next=NULL;
40     temp=head;
41     temp=temp->next;
42     temp->prev=NULL;
43     head=temp;
44     printf("After deleting at beginning : \n");
45     while(head!=NULL)
46     {
47         printf("%d->",head->data);
48         head=head->next;
49     }
50     printf("NULL\n");
51     return 0;
52 }
```

### Output:

```
input
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:

```
main.c
1 //S.PRAVEEN KUMAR
2 //ch.en.u4aie22048 AIE
3 //Lab-4
4
5 //deleting in end in double Linked list
6 #include<stdio.h>
7 #include<stdlib.h>
8 struct node{
9     int data;
10    struct node* next;
11    struct node* prev;
12 };
13 int main()
14 {
15     int num,i,number,len;
16     struct node* head;
17     struct node* temp;
18     struct node* temp2;
19     struct node* newnode;
20     head=(struct node*)malloc(sizeof(struct node));
21     temp=malloc(sizeof(struct node));
22     temp=head;
23     printf("Enter the size in linked list: \n");
24     scanf("%d",&num);
25     temp->prev=NULL;
26     newnode=malloc(sizeof(struct node));
27     for(i=0;i<num;i++)
28     {
29         printf("element %d: ",i);
30         scanf("%d",&number);
31         temp->data=number;
32         if(i!=num-1)
33         {
34             temp->next=malloc(sizeof(struct node));
35             newnode=temp;
36             temp=temp->next;
37             temp->prev=newnode;
38         }
39     }
40     temp->next=NULL;
41     temp=temp->prev;
42     temp->next=NULL;
43     temp2=temp->prev;
44     temp->prev=temp2;
45     printf("After deleting at end : \n");
46     while(head!=NULL)
47     {
48         printf("%d->",head->data);
49         head=head->next;
50     }
51     printf("NULL\n");
52     return 0;
53 }
```

### Output:

```
input
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:

```
main.c
1 //S.PRAVEEN KUMAR
2 //ch.en.u4aie22048 AIE
3 //Lab-4
4
5 //deleting in specific position in double linked list
6 #include<stdio.h>
7 #include<stdlib.h>
8 struct node{
9     int data;
10    struct node* next;
11    struct node* prev;
12};
13 int main()
14 {
15     int num,i,number,len,newnum,newpos;
16     struct node* head,*temp;
17     head=(struct node*)malloc(sizeof(struct node));
18     temp=malloc(sizeof(struct node));
19     temp=head;
20     printf("Deleting at specific postion\n");
21     printf("Enter the size in Double linked list: ");
22     scanf("%d",&num);
23     temp->prev=NULL;
24     for(i=0;i<num;i++)
25     {
26         printf("element %d: ",i);
27         scanf("%d",&number);
28         temp->data=number;
29         if(i!=num-1)
30         {
31             temp->next=malloc(sizeof(struct node));
32             temp->next->prev=temp;
33             temp=temp->next;
34         }
35     }
36     temp->next=NULL;
37     printf("Enter the Postion to delete: ");
38     scanf("%d",&newpos);
39     temp=head;
40
41     for(i=0;i<newpos-1;i++)
42     {
43         temp=temp->next;
44     }
45     temp->next=temp->next->next;
46     temp->next->prev=temp;
47     while(head!=NULL)
48     {
49         printf("%d->",head->data);
50         head=head->next;
51     }
52     printf("NULL\n");
53     return 0;
54 }
```

### Output:

```
input
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.
```