

VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES - TECHNICAL CAMPUS

Grade **A++** Accredited Institution by NAAC

NBA Accredited for MCA Programme; Recognized under Section 2(f) by UGC;
Affiliated to GGSIP University, Delhi; Recognized by Bar Council of India and AICTE

An ISO 9001:2015 Certified Institution

SCHOOL OF ENGINEERING & TECHNOLOGY

B. Tech Programme: B. Tech AI-ML (A)

Course Title: Data Structures Lab

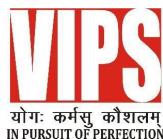
Course Code: AIML-201

Submitted To:

Ms. Nishi Jain
Assistant Professor

Submitted By:

Name: Kunsh Sabharwal
Enrolment No: 01117711623



VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES - TECHNICAL CAMPUS

Grade A++ Accredited Institution by NAAC

NBA Accredited for MCA Programme; Recognized under Section 2(f) by UGC;
Affiliated to GGSIP University, Delhi; Recognized by Bar Council of India and AICTE

An ISO 9001:2015 Certified Institution

SCHOOL OF ENGINEERING & TECHNOLOGY

VISION OF INSTITUTE

To be an educational institute that empowers the field of engineering to build a sustainable future by providing quality education with innovative practices that supports people, planet and profit.

MISSION OF INSTITUTE

To groom the future engineers by providing value-based education and awakening students' curiosity, nurturing creativity and building capabilities to enable them to make significant contributions to the world.



IN PURSUIT OF PERFECTION

Grade A++ Accredited Institution by NAAC

NBA Accredited for MCA Programme; Recognized under Section 2(f) by UGC;
Affiliated to GGSIP University, Delhi; Recognized by Bar Council of India and AICTE

SCHOOL OF ENGINEERING & TECHNOLOGY

INDEX



यागः कमसु काशलम् IN PURSUIT OF PERFECTION

IN PURSUIT OF PERFECTION

Grade A++ Accredited Institution by NAAC

NBA Accredited for MCA Programme: Recognized under Section 2(f) by UGC:

NBA Accredited for MBA Programme, Recognized under Section 2(f) by UGC,
Affiliated to GGSIP University, Delhi: Recognized by Bar Council of India and AICTE

An ISO 9001:2015 Certified Institution

AN ISO 9001:2015 Certified Institution
SCHOOL OF ENGINEERING & TECHNOLOGY



यागः कमसु काशलम् IN PURSUIT OF PERFECTION

IN PURSUIT OF PERFECTION

Grade A++ Accredited Institution by NAAC

NBA Accredited for MCA Programme; Recognized under Section 2(f) by UGC;

NBA Accredited for MBA Programme, Recognized under Section 2(f) by UGC,
Affiliated to GGSIP University, Delhi: Recognized by Bar Council of India and AICTE

An ISO 9001:2015 Certified Institution

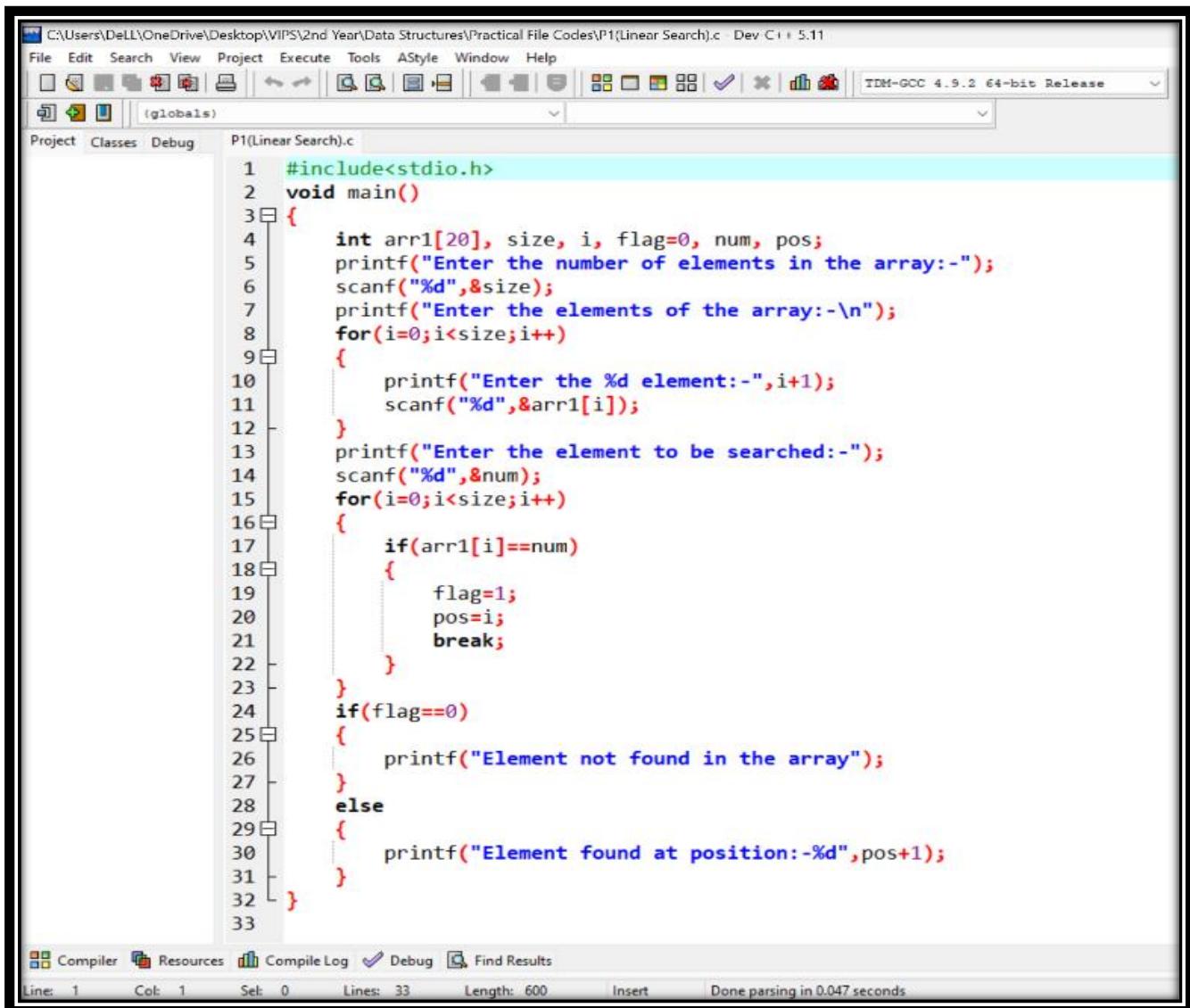
AN ISO 9001:2015 Certified Institution
SCHOOL OF ENGINEERING & TECHNOLOGY

EXPERIMENT 1(a)

Problem statement: Write a program to implement linear search.

Algorithm:

Programming Code:



The screenshot shows the Dev C++ IDE interface with the following details:

- Title Bar:** C:\Users\DELL\OneDrive\Desktop\VIIPS\2nd Year\Data Structures\Practical File Codes\P1\Linear Search.c Dev C++ 5.11
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help
- Toolbar:** Includes icons for New, Open, Save, Print, Cut, Copy, Paste, Find, Replace, etc.
- Project Explorer:** Shows "globals" under "Project".
- Code Editor:** Displays the C code for linear search. The code uses standard input-output functions and a for loop to iterate through the array elements.
- Status Bar:** Shows "Line: 1 Col: 1 Sel: 0 Lines: 33 Length: 600 Insert Done parsing in 0.047 seconds".

```
#include<stdio.h>
void main()
{
    int arr1[20], size, i, flag=0, num, pos;
    printf("Enter the number of elements in the array:-");
    scanf("%d",&size);
    printf("Enter the elements of the array:-\n");
    for(i=0;i<size;i++)
    {
        printf("Enter the %d element:-",i+1);
        scanf("%d",&arr1[i]);
    }
    printf("Enter the element to be searched:-");
    scanf("%d",&num);
    for(i=0;i<size;i++)
    {
        if(arr1[i]==num)
        {
            flag=1;
            pos=i;
            break;
        }
    }
    if(flag==0)
    {
        printf("Element not found in the array");
    }
    else
    {
        printf("Element found at position:-%d",pos+1);
    }
}
```

Output:

```
C:\Users\DeLL\OneDrive\Desktop X + ▾
Enter the number of elements in the array:-5
Enter the elements of the array:-
Enter the 1 element:-1
Enter the 2 element:-2
Enter the 3 element:-3
Enter the 4 element:-4
Enter the 5 element:-5
Enter the element to be searched:-3
Element found at position:-3
-----
Process exited after 9.368 seconds with return value 28
Press any key to continue . . . |
```

Learning Outcomes:

EXPERIMENT 1(b)

Problem statement: Write a program to implement Binary search.

Algorithm:

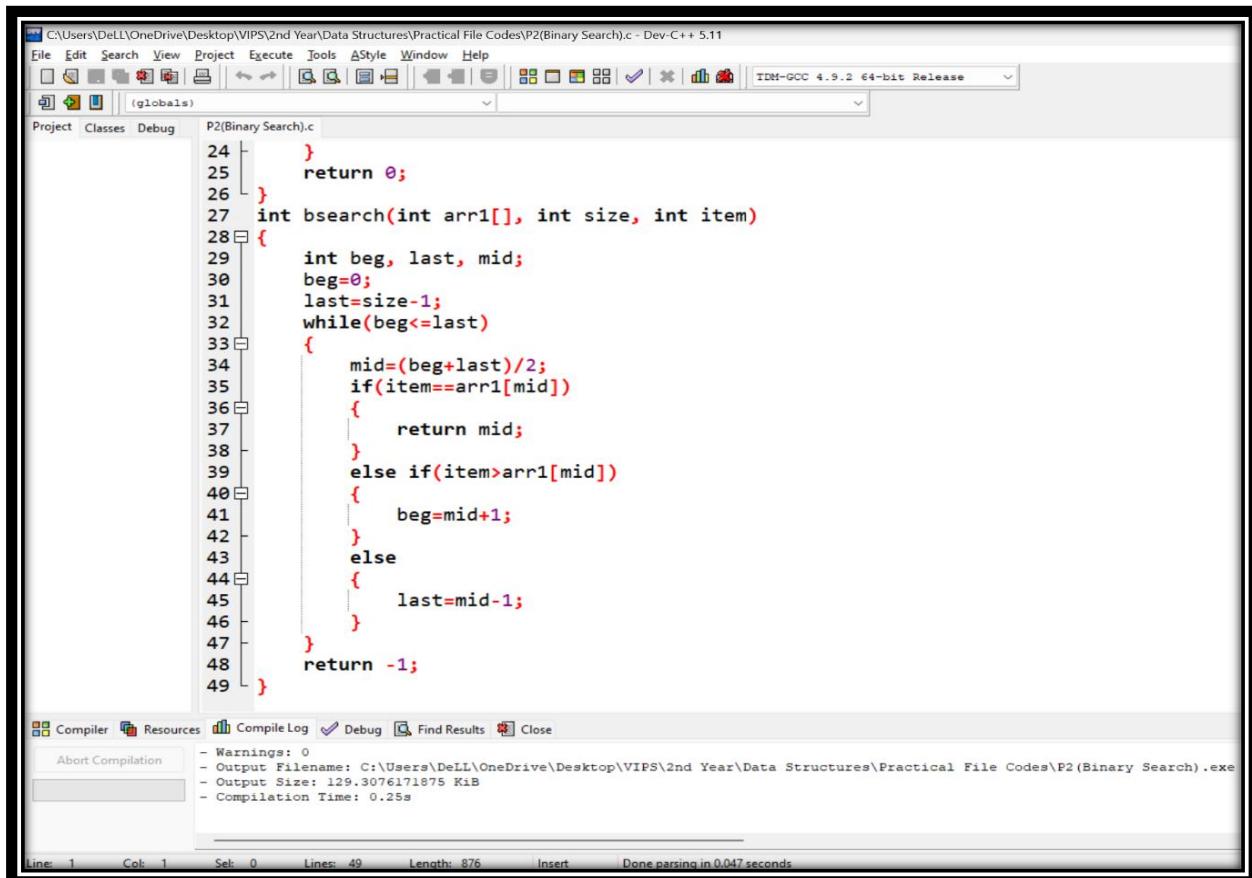
Programming Code:

The screenshot shows the Dev-C++ IDE interface with the following details:

- Title Bar:** C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P2(Binary Search).c - Dev-C++ 5.11
- Toolbar:** Standard Dev-C++ toolbar with icons for file operations, project management, and compilation.
- MenuBar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help.
- ToolBox:** Standard Dev-C++ tool palette.
- Project Bar:** Shows the current project is P2(Binary Search).
- Code Editor:** The main window displays the C code for a binary search program. The code includes a main function that prompts the user for array size and elements, then calls a bsearch function to find a specific item.

```
1 #include<stdio.h>
2 int bsearch(int [], int, int);
3 int main()
4 {
5     int arr1[20], item, size, index, i;
6     printf("Enter the number of elements in the array:-");
7     scanf("%d",&size);
8     printf("Enter the array elements in ascending order:-\n");
9     for(i=0;i<size;i++)
10    {
11        printf("Enter the %d element:-",i+1);
12        scanf("%d",&arr1[i]);
13    }
14    printf("Enter the element to be searched:-");
15    scanf("%d",&item);
16    index=bsearch(arr1,size,item);
17    if(index==-1)
18    {
19        printf("Element was not found");
20    }
21    else
22    {
23        printf("Element found at position:-%d",index+1);
24    }
25    return 0;
26 }
27 int bsearch(int arr1[], int size, int item)
```

- Compiler Tab:** Shows compilation results with 0 warnings and 0 errors.
- Resources Tab:** Shows output filename (C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P2(Binary Search).exe), output size (129.3076171875 KiB), and compilation time (0.25s).
- Status Bar:** Displays line (Line: 1), column (Col: 1), selection (Sel: 0), lines (Lines: 49), length (Length: 876), and message (Done parsing in 0.047 seconds).



The screenshot shows the Dev-C++ IDE interface. The title bar reads "C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P2(Binary Search).c - Dev-C++ 5.11". The menu bar includes File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, and Help. The toolbar has various icons for file operations like Open, Save, Print, etc. The status bar at the bottom shows "Line: 1 Col: 1 Sel: 0 Lines: 49 Length: 876 Insert Done parsing in 0.047 seconds".

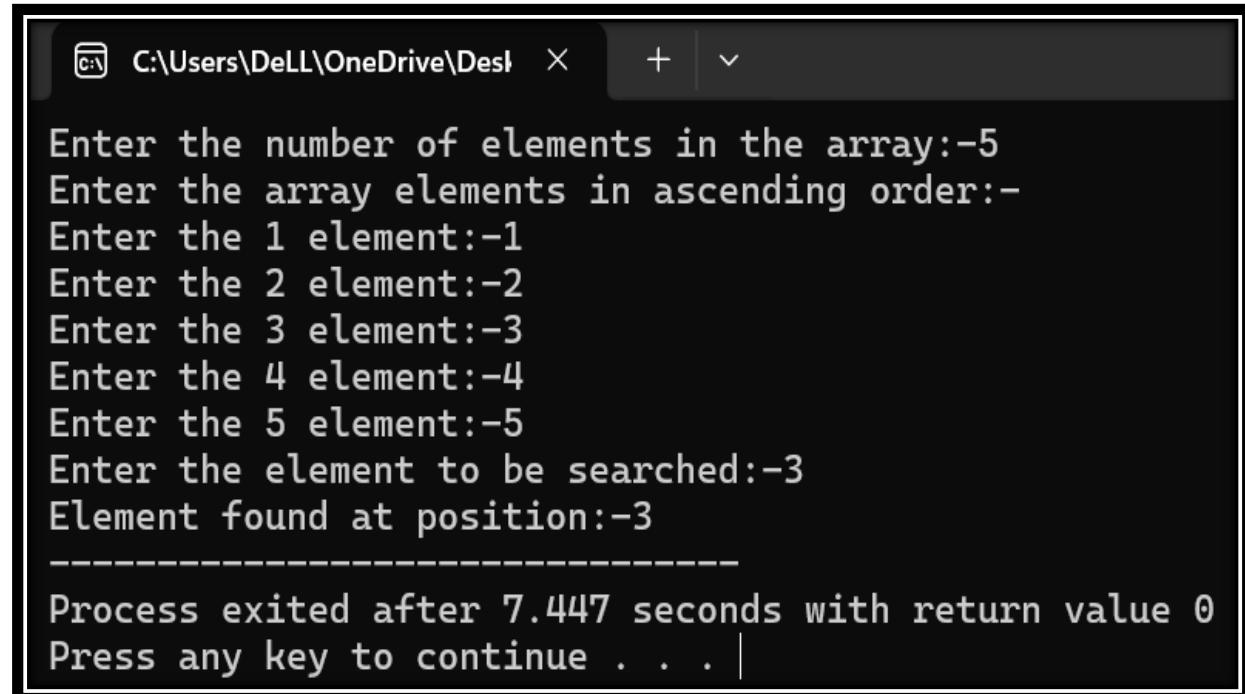
```

24     }
25     return 0;
26 }
27 int bsearch(int arr1[], int size, int item)
28 {
29     int beg, last, mid;
30     beg=0;
31     last=size-1;
32     while(beg<=last)
33     {
34         mid=(beg+last)/2;
35         if(item==arr1[mid])
36         {
37             return mid;
38         }
39         else if(item>arr1[mid])
40         {
41             beg=mid+1;
42         }
43         else
44         {
45             last=mid-1;
46         }
47     }
48     return -1;
49 }

```

The code implements a binary search algorithm. It takes an array, its size, and a target item as input. It initializes beg to 0 and last to size-1. It then enters a loop where it calculates the middle index mid as (beg+last)/2. If the item at index mid is equal to the target, it returns mid. If the item is greater than the item at index mid, it updates beg to mid+1. Otherwise, it updates last to mid-1. If the loop exits without finding the item, it returns -1.

Output:



The terminal window shows the following interaction:

```

Enter the number of elements in the array:-5
Enter the array elements in ascending order:-
Enter the 1 element:-1
Enter the 2 element:-2
Enter the 3 element:-3
Enter the 4 element:-4
Enter the 5 element:-5
Enter the element to be searched:-3
Element found at position:-3
-----
Process exited after 7.447 seconds with return value 0
Press any key to continue . . .

```

The user enters 5 as the number of elements. Then they enter the array elements in ascending order: -1, -2, -3, -4, -5. They then enter -3 as the element to be searched. The program outputs that the element was found at position -3. Finally, it exits after 7.447 seconds with a return value of 0.

Learning Outcomes:

EXPERIMENT 2

Problem statement: Create a stack and perform Peek, Pop, Push and Traverse operations on the stack using an array.

Algorithm:

Programming Code:

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3(Stack Operations).c - Dev-C++ 5.11

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 int stack[100],i,j,choice=0, n, top=-1, i, x, y;
4 int peek();
5 int pop();
6 int push();
7 int traverse();
8 int main()
9 {
10    printf("*****Stack Opearations using array*****\n");
11    printf("Enter the number of elements in the stack:-");
12    scanf("%d",&n);
13    for(i=0;i<n;i++)
14    {
15        int a;
16        printf("Enter the %d element:-",i+1);
17        scanf("%d",&a);
18        stack[i]=a;
19    }
20
21

```

Compiler Resources Compile Log Debug Find Results Close

- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3(Stack Operations).exe
- Output Size: 130.67578125 KiB
- Compilation Time: 0.22s

Line: 1 Col: 1 Sel: 0 Lines: 118 Length: 2304 Insert Done parsing in 0.078 seconds

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3(Stack Operations).c - Dev-C++ 5.11

```

22    printf("\n-----\n");
23    top=n-1;
24    while(choice!=5)
25    {
26        printf("\n");
27        printf("Choose one from the below operations:-\n");
28        printf("\n1.Peek\n2.Pop\n3.Push\n4.Traverse\n5.Exit\n");
29        printf("Enter your choice:-");
30        scanf("%d",&choice);
31        printf("\n");
32
33        switch(choice)
34        {
35            case 1:
36                peek();
37                break;
38            case 2:
39                pop();
40                break;
41            case 3:
42                push();

```

Compiler Resources Compile Log Debug Find Results Close

- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3(Stack Operations).exe
- Output Size: 130.67578125 KiB
- Compilation Time: 0.22s

Line: 1 Col: 1 Sel: 0 Lines: 118 Length: 2304 Insert Done parsing in 0.078 seconds

KUNSH SABHARWAL

The screenshot shows the Dev-C++ IDE interface with the file `P3(Stack Operations).c` open. The code implements stack operations using a global array `stack`. It includes functions for pushing values onto the stack and popping values from it. The code is annotated with comments explaining its functionality.

```
40     break;
41 case 3:
42     push();
43     break;
44 case 4:
45     traverse();
46     break;
47 case 5:
48     printf("\n");
49     printf("Thanks for using the code");
50     exit(0);
51     break;
52 default:
53     printf("Invalid operation\n");
54     break;
55 }
56 }
57 return 0;
58 }
59 int push()
60 {
61     int val;
62     if(top==n)
```

Compiler Log:

- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIIPS\2nd Year\Data Structures\Practical File Codes\P3(Stack Operations).exe
- Output Size: 130.67578125 KiB
- Compilation Time: 0.22s

Line: 1 Col: 1 Sel: 0 Lines: 118 Length: 2304 Insert Done parsing in 0.078 seconds

The screenshot shows the Dev-C++ IDE interface with the file `P3(Stack Operations).c` open. The code implements stack operations using a global array `stack`. It includes functions for pushing values onto the stack and popping values from it. The code is annotated with comments explaining its functionality.

```
61     int val;
62     if(top==n)
63     {
64         printf("\nOverflow");
65     }
66     else
67     {
68         printf("Enter the value:-");
69         scanf("%d",&val);
70         top=top+1;
71         stack[top]=val;
72     }
73     return 0;
74 }
75 int pop()
76 {
77     if(top==-1)
78     {
79         printf("\nUnderflow");
80     }
81     else
82     {
83         v=stack[top];
84         top=top-1;
85     }
86     return v;
87 }
```

Compiler Log:

- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIIPS\2nd Year\Data Structures\Practical File Codes\P3(Stack Operations).exe
- Output Size: 130.67578125 KiB
- Compilation Time: 0.22s

Line: 1 Col: 1 Sel: 0 Lines: 118 Length: 2304 Insert Done parsing in 0.078 seconds

KUNSH SABHARWAL

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3(Stack Operations).c - Dev-C++ 5.11

```
79         printf("\nUnderflow");
80     }
81     else
82     {
83         y=stack[top];
84         printf("The popped element is:-%d\n",y);
85         top=top-1;
86     }
87     return 0;
88 }
89 int peek()
90 {
91     if(top!=-1)
92     {
93         printf("The element at the top is:-%d\n",stack[top]);
94     }
95     else
96     {
97         printf("\nUnderflow\n");
98     }
99     return 0;
100 }
101 int traverse()
```

Compiler Resources Compile Log Debug Find Results Close

- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3(Stack Operations).exe
- Output Size: 130.67578125 KiB
- Compilation Time: 0.22s

Line: 1 Col: 1 Sel: 0 Lines: 118 Length: 2304 Insert Done parsing in 0.078 seconds

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3(Stack Operations).c - Dev-C++ 5.11

```
97         printf("\nUnderflow\n");
98     }
99     return 0;
100 }
101 int traverse()
102 {
103     if(top==-1)
104     {
105         printf("\nUnderflow\n");
106     }
107     else
108     {
109         printf("The values in the stack are:-\n");
110         for(x=top;x>=0;x--)
111         {
112             printf("%d\n",stack[x]);
113         }
114     }
115     printf("\n");
116     return 0;
117 }
```

Compiler Resources Compile Log Debug Find Results Close

- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3(Stack Operations).exe
- Output Size: 130.67578125 KiB
- Compilation Time: 0.22s

Line: 1 Col: 1 Sel: 0 Lines: 118 Length: 2304 Insert Done parsing in 0.078 seconds

Output:

```
*****Stack Operations using array*****
Enter the number of elements in the stack:-5
Enter the 1 element:-5
Enter the 2 element:-10
Enter the 3 element:-15
Enter the 4 element:-20
Enter the 5 element:-25

-----
Choose one from the below operations:-
1.Peek
2.Pop
3.Push
4.Traverse
5.Exit
Enter your choice:-1

The element at the top is:-25

Choose one from the below operations:-
1.Peek
2.Pop
3.Push
4.Traverse
5.Exit
Enter your choice:-2

The popped element is:-25

Choose one from the below operations:-
1.Peek
2.Pop
3.Push
4.Traverse
5.Exit
Enter your choice:-1

The element at the top is:-20

Choose one from the below operations:-
```

```
The element at the top is:-20
Choose one from the below operations:-
1.Peek
2.Pop
3.Push
4.Traverse
5.Exit
Enter your choice:-3

Enter the value:-25

Choose one from the below operations:-
1.Peek
2.Pop
3.Push
4.Traverse
5.Exit
Enter your choice:-1

The element at the top is:-25

Choose one from the below operations:-
1.Peek
2.Pop
3.Push
4.Traverse
5.Exit
Enter your choice:-4

The values in the stack are:-
25
20
15
10
5

Choose one from the below operations:-
1.Peek
2.Pop
3.Push
4.Traverse
5.Exit
Enter your choice:-5

Thanks for using the code
-----
Process exited after 28.24 seconds with return value 0
Press any key to continue . . . |
```

Learning Outcome:

EXPERIMENT 3(a)

Problem statement: Write programs to implement Bubble Sort and Insertion Sort.

Algorithm:

Programming Code:

The screenshot shows the Dev-C++ IDE interface with the file `P3a(Bubble Sort).c` open. The code implements a simple bubble sort algorithm. It prompts the user for the number of elements and each element's value. It then prints the array before sorting and again after sorting. The code uses nested loops to compare adjacent elements and swap them if they are in the wrong order.

```

1 #include<stdio.h>
2 int BubbleSort(int arr1[], int n);
3 void main()
4 {
5     int n, i;
6     printf("Enter the number of elements in the array:-");
7     scanf("%d",&n);
8     int a[n];
9     for(i=0;i<n;i++)
10    {
11        printf("Enter the %d element:-",i+1);
12        scanf("%d",&a[i]);
13    }
14    printf("Array before Sorting is:-");
15    for(i=0;i<n;i++)
16    {
17        printf("%d\t",a[i]);
18    }
19    BubbleSort(a,n);
20    printf("\nSorted Array is:-");
21    for(i=0;i<n;i++)
22    {
23        printf("%d\t",a[i]);
24    }
25 }
26 int BubbleSort(int arr1[], int n)

```

Compiler Log:

- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3a(Bubble Sort).exe
- Output Size: 129.798828125 KiB
- Compilation Time: 0.22s

The screenshot shows the Dev-C++ IDE interface with the file `P3a(Bubble Sort).c` open. This version of the code is more optimized. It includes an inner loop that only runs up to $n-i-1$ because after i iterations, the last i elements are already sorted. It also includes a check to see if the array is already sorted, which can break the loop early if it finds no swaps.

```

25 L}
26 int BubbleSort(int arr1[], int n)
27 {
28     int temp, i, j;
29     for(i=0;i<=n-2;i++)
30    {
31        int c=0;
32        for(j=0;j<=n-i-2;j++)
33        {
34            if(arr1[j]>arr1[j+1])
35            {
36                temp=arr1[j];
37                arr1[j]=arr1[j+1];
38                arr1[j+1]=temp;
39                c=1;
40            }
41        }
42        if(c==0)
43        {
44            printf("Array is already sorted\n");
45            break;
46        }
47    }
48    return 0;
49 }

```

Compiler Log:

- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3a(Bubble Sort).exe
- Output Size: 129.798828125 KiB
- Compilation Time: 0.22s

KUNSH SABHARWAL

The screenshot shows the Dev-C++ IDE interface with the file `P3a(Insertion Sort).c` open. The code implements an insertion sort algorithm. The `main()` function prompts the user for the number of elements and the elements themselves, then prints the array before sorting. It calls the `InsertionSort` function, which then prints the sorted array. The `InsertionSort` function uses a nested loop to shift elements and insert the current element at the correct position.

```
1 #include<stdio.h>
2 void main()
3 {
4     int n, i;
5     printf("Enter the number of elements in the array:-");
6     scanf("%d",&n);
7     int a[n];
8     for(i=0;i<n;i++)
9     {
10         printf("Enter the %d element:-",i+1);
11         scanf("%d",&a[i]);
12     }
13     printf("Array before Sorting is:-");
14     for(i=0;i<n;i++)
15     {
16         printf("%d\t",a[i]);
17     }
18     InsertionSort(a,n);
19     printf("\nSorted Array is:-");
20     for(i=0;i<n;i++)
21     {
22         printf("%d\t",a[i]);
23     }
24 }
25 int InsertionSort(int arr1[], int n)
26 {
27 }
```

Compiler Log:

- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3a(Insertion Sort).exe
- Output Size: 129.654296875 KiB
- Compilation Time: 0.19s

The screenshot shows the Dev-C++ IDE interface with the file `P3a(Insertion Sort).c` open. The code implements an insertion sort algorithm. The `main()` function calls the `InsertionSort` function, which then prints the sorted array. The `InsertionSort` function uses a nested loop to shift elements and insert the current element at the correct position.

```
17 }
18 InsertionSort(a,n);
19 printf("\nSorted Array is:-");
20 for(i=0;i<n;i++)
21 {
22     printf("%d\t",a[i]);
23 }
24 }
25 int InsertionSort(int arr1[], int n)
26 {
27     int temp, i, c;
28     for(i=1;i<=n;i++)
29     {
30         int j=i-1;
31         temp=arr1[i];
32         while (j>=0 && arr1[j]>temp)
33         {
34             arr1[j+1]=arr1[j];
35             j=j-1;
36         }
37         arr1[j+1]=temp;
38     }
39     return 0;
40 }
41 |
```

Compiler Log:

- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3a(Insertion Sort).exe
- Output Size: 129.654296875 KiB
- Compilation Time: 0.19s

Output:

```
C:\Users\DeLL\OneDrive\Desktop + ▾  
Enter the number of elements in the array:-5  
Enter the 1 element:-55  
Enter the 2 element:-12  
Enter the 3 element:-62  
Enter the 4 element:-91  
Enter the 5 element:-5  
Array before Sorting is:-55      12      62      91      5  
Sorted Array is:-5      12      55      62      91  
-----  
Process exited after 10.15 seconds with return value 5  
Press any key to continue . . . |
```

```
C:\Users\DeLL\OneDrive\Desktop + ▾  
Enter the number of elements in the array:-5  
Enter the 1 element:-7  
Enter the 2 element:-19  
Enter the 3 element:-5  
Enter the 4 element:-22  
Enter the 5 element:-1  
Array before Sorting is:-7      19      5      22      1  
Sorted Array is:-1      5      7      19      22  
-----  
Process exited after 13.21 seconds with return value 5  
Press any key to continue . . . |
```

Learning Outcome:

EXPERIMENT 3(b)

Problem statement: Write programs to implement Selection Sort and Merge Sort.

Algorithm:

Programming Code:

The screenshot shows the Dev-C++ IDE interface with the following details:

- Title Bar:** C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3b(Selection Sort).c - Dev-C++ 5.11
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help
- Toolbar:** Standard Dev-C++ toolbar.
- Project Tab:** Project, Classes, Debug (selected).
- Code Editor:**

```

1 #include<stdio.h>
2 void Selection(int arr[], int n);
3 void main()
4 {
5     int n, i;
6     printf("Enter the number of elements in the array:-");
7     scanf("%d",&n);
8     int a[n];
9     for(i=0;i<n;i++)
10    {
11        printf("Enter the %d element:-",i+1);
12        scanf("%d",&a[i]);
13    }
14    printf("Array before Sorting is:-");
15    for(i=0;i<n;i++)
16    {
17        printf("%d\t",a[i]);
18    }
19    Selection(a,n);
20    printf("\nSorted Array is:-");
21    for(i=0;i<n;i++)
22    {
23        printf("%d\t",a[i]);
24    }
25 }
26 void Selection(int arr[], int n)

```
- Compiler Tab:** Compiler, Resources, Compile Log, Debug, Find Results, Close.
- Compile Log:**
 - Warnings: 0
 - Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3b(Selection Sort).exe
 - Output Size: 129.650390625 KiB
 - Compilation Time: 0.27s
- Status Bar:** Line: 39, Col: 12, Sel: 0, Lines: 41, Length: 688, Insert, Done parsing in 0.015 seconds.

The screenshot shows the Dev-C++ IDE interface with the following details:

- Title Bar:** C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3b(Selection Sort).c - Dev-C++ 5.11
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help
- Toolbar:** Standard Dev-C++ toolbar.
- Project Tab:** Project, Classes, Debug (selected).
- Code Editor:**

```

17     printf("%d\t",a[i]);
18 }
19 Selection(a,n);
20 printf("\nSorted Array is:-");
21 for(i=0;i<n;i++)
22 {
23     printf("%d\t",a[i]);
24 }
25 }
26 void Selection(int arr[], int n)
27 {
28     int i, j, small;
29     for (i=0;i<n-1;i++)
30     {
31         small=i;
32         for (j=i+1;j<n;j++)
33             if (arr[j]<arr[small])
34             {
35                 small=j;
36             }
37         int temp=arr[small];
38         arr[small]=arr[i];
39         arr[i]=temp;
40     }
41 }

```
- Compiler Tab:** Compiler, Resources, Compile Log, Debug, Find Results, Close.
- Compile Log:**
 - Warnings: 0
 - Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3b(Selection Sort).exe
 - Output Size: 129.650390625 KiB
 - Compilation Time: 0.27s
- Status Bar:** Line: 49, Col: 20, Sel: 0, Lines: 41, Length: 688, Insert, Done parsing in 0.015 seconds.

KUNSH SABHARWAL

The screenshot shows the Dev-C++ IDE interface with the code editor displaying the `P3b(Merge Sort).c` file. The code implements the `mergeArrays` function using two temporary arrays, `l_array` and `r_array`, to merge two sorted arrays `arr` from index `p` to `q` and `q` to `r`. The merged result is stored back into `arr`. The code uses nested loops to iterate through the arrays and a `while` loop to handle the remaining elements after the merge point.

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 void mergeArrays(int arr[], int p, int q, int r)
4 {
5     int i, j;
6     int n1 = q - p + 1;
7     int n2 = r - q;
8     int *l_array = (int *)malloc(sizeof(int) * n1);
9     int *r_array = (int *)malloc(sizeof(int) * n2);
10    for (i = 0; i < n1; i++)
11        l_array[i] = arr[p + i];
12    for (j = 0; j < n2; j++)
13        r_array[j] = arr[q + 1 + j];
14    i = 0;
15    j = 0;
16    int k = p;
17    while (i < n1 && j < n2)
18    {
19        if (l_array[i] <= r_array[j])
20        {
21            arr[k] = l_array[i];
22            i++;
23        }
24        else
25        {
26            arr[k] = r_array[j];
27            j++;
28        }
29        k++;
30    }
31    while (i < n1)
32    {
33        arr[k] = l_array[i];
34        i++;
35        k++;
36    }
37    while (j < n2)
38    {
39        arr[k] = r_array[j];
40        j++;
41        k++;
42    }
43    free(l_array);
44    l_array = NULL;
45
46    free(r_array);
47    r_array = NULL;
48 }
49 void ms(int arr[], int p, int r)
50 {
51     if (p >= r)
52         return;
53 }
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3b(Merge Sort).exe
- Output Size: 130.1767578125 Kib
- Compilation Time: 0.28s

Line: 40 Col: 13 Sel: 0 Lines: 87 Length: 1621 Insert Done parsing in 0 seconds

The screenshot shows the Dev-C++ IDE interface with the code editor displaying the `P3b(Merge Sort).c` file. The code implements the `ms` function, which performs a merge sort on the array `arr` from index `p` to `r`. It uses the `mergeArrays` function to merge two halves of the array. The base case for the recursion is when `p >= r`, in which case it returns immediately. The code uses standard C memory management (`malloc` and `free`) for temporary arrays.

```
1 #include<stdio.h>
2 #include<stdlib.h>
3 void mergeArrays(int arr[], int p, int q, int r)
4 {
5     int i, j;
6     int n1 = q - p + 1;
7     int n2 = r - q;
8     int *l_array = (int *)malloc(sizeof(int) * n1);
9     int *r_array = (int *)malloc(sizeof(int) * n2);
10    for (i = 0; i < n1; i++)
11        l_array[i] = arr[p + i];
12    for (j = 0; j < n2; j++)
13        r_array[j] = arr[q + 1 + j];
14    i = 0;
15    j = 0;
16    int k = p;
17    while (i < n1 && j < n2)
18    {
19        if (l_array[i] <= r_array[j])
20        {
21            arr[k] = l_array[i];
22            i++;
23        }
24        else
25        {
26            arr[k] = r_array[j];
27            j++;
28        }
29        k++;
30    }
31    while (i < n1)
32    {
33        arr[k] = l_array[i];
34        i++;
35        k++;
36    }
37    while (j < n2)
38    {
39        arr[k] = r_array[j];
40        j++;
41        k++;
42    }
43    free(l_array);
44    l_array = NULL;
45
46    free(r_array);
47    r_array = NULL;
48 }
49 void ms(int arr[], int p, int r)
50 {
51     if (p >= r)
52         return;
53 }
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3b(Merge Sort).exe
- Output Size: 130.1767578125 Kib
- Compilation Time: 0.28s

Line: 23 Col: 10 Sel: 0 Lines: 87 Length: 1621 Insert Done parsing in 0 seconds

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P3b(Merge Sort).c - Dev-C++ 5.11

File Edit Search View Project Execute Tools Style Window Help

Project Classes Debug [*] P3b(Merge Sort).c P3a(Insertion Sort).c

```
53     int q = p + (r - p) / 2;
54     ms(arr, p, q);
55     ms(arr, q + 1, r);
56
57     mergeArrays(arr, p, q, r);
58 }
59 void merge_sort(int arr[], int n)
60 {
61     ms(arr, 0, n - 1);
62 }
63 void main()
64 {
65     int n, i;
66     printf("Enter the number of elements in the array:-");
67     scanf("%d", &n);
68     int arr[n];
69     for(i=0;i<n;i++)
70     {
71         printf("Enter the %d element:-", i+1);
72         scanf("%d", &arr[i]);
73     }
74     printf("Array before Sorting is:-");
75     for(i=0;i<n;i++)
76     {
77         printf("%d\t", arr[i]);
78     }
79     merge_sort(arr, n);
80     printf("\nSorted Array is:-");
81     for(i=0;i<n;i++)
82     {
83         printf("%d\t", arr[i]);
84     }
85 }
```

Line: 52 Col: 16 Sel: 0 Lines: 85 Length: 1617 Insert Done parsing in 0 seconds

Outputs:

```
C:\Users\DeLL\OneDrive\Desktop + 
Enter the number of elements in the array:-5
Enter the 1 element:-5
Enter the 2 element:-14
Enter the 3 element:-12
Enter the 4 element:-9
Enter the 5 element:-8
Array before Sorting is:-5      14      12      9      8
Sorted Array is:-5      8      9      12      14
-----
Process exited after 14.29 seconds with return value 5
Press any key to continue . . . |
```

```
C:\Users\DeLL\OneDrive\Desktop + 
Enter the number of elements in the array:-5
Enter the 1 element:-1
Enter the 2 element:-6
Enter the 3 element:-2
Enter the 4 element:-5
Enter the 5 element:-10
Array before Sorting is:-1      6      2      5      10
Sorted Array is:-1      2      5      6      10
-----
Process exited after 9.383 seconds with return value 5
Press any key to continue . . . |
```

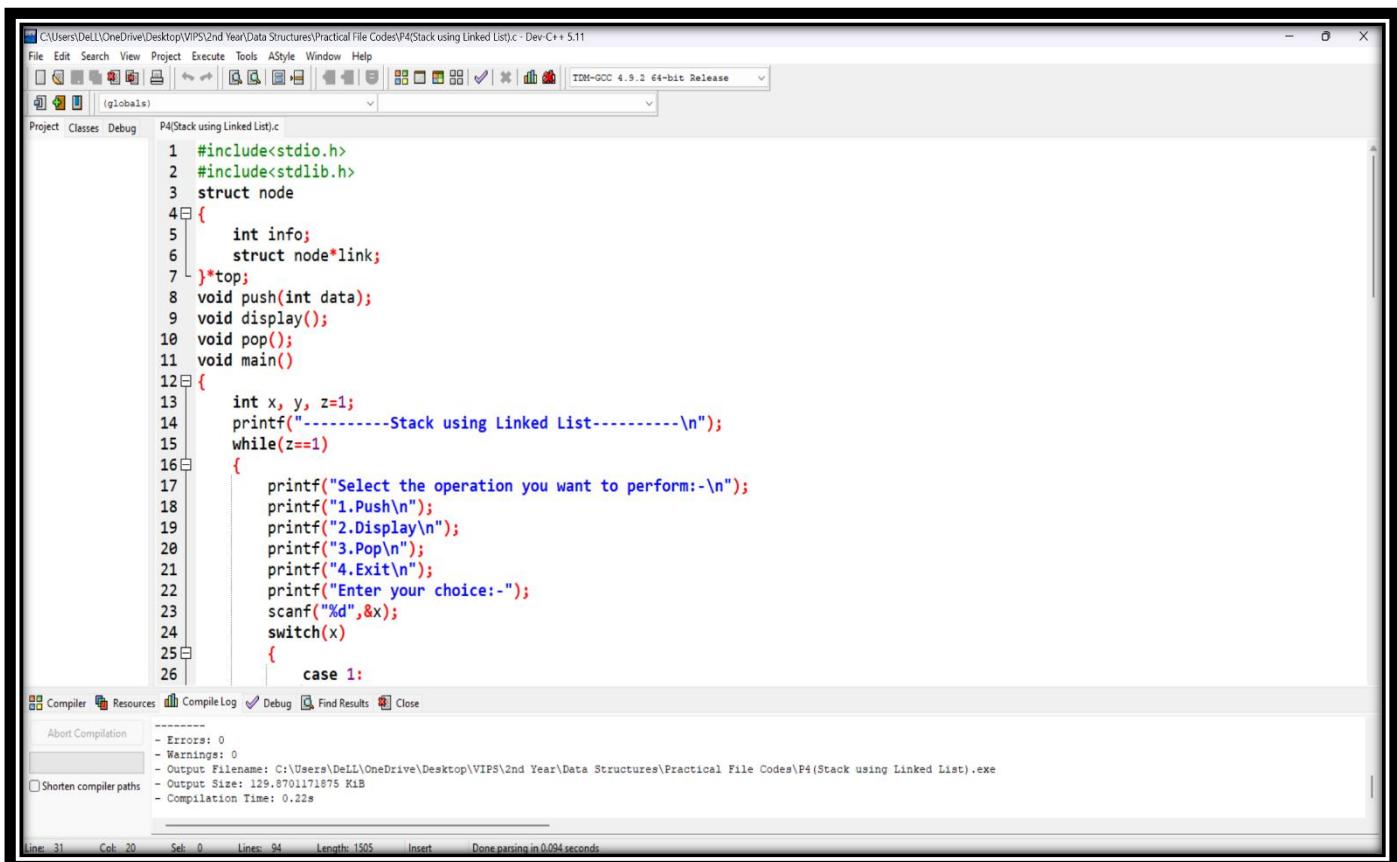
Learning Outcome:

EXPERIMENT 4

Problem statement: Write a program to implement a Linked List using Stack and perform operations such as push, pop and display the list.

Algorithm:

Source Code:



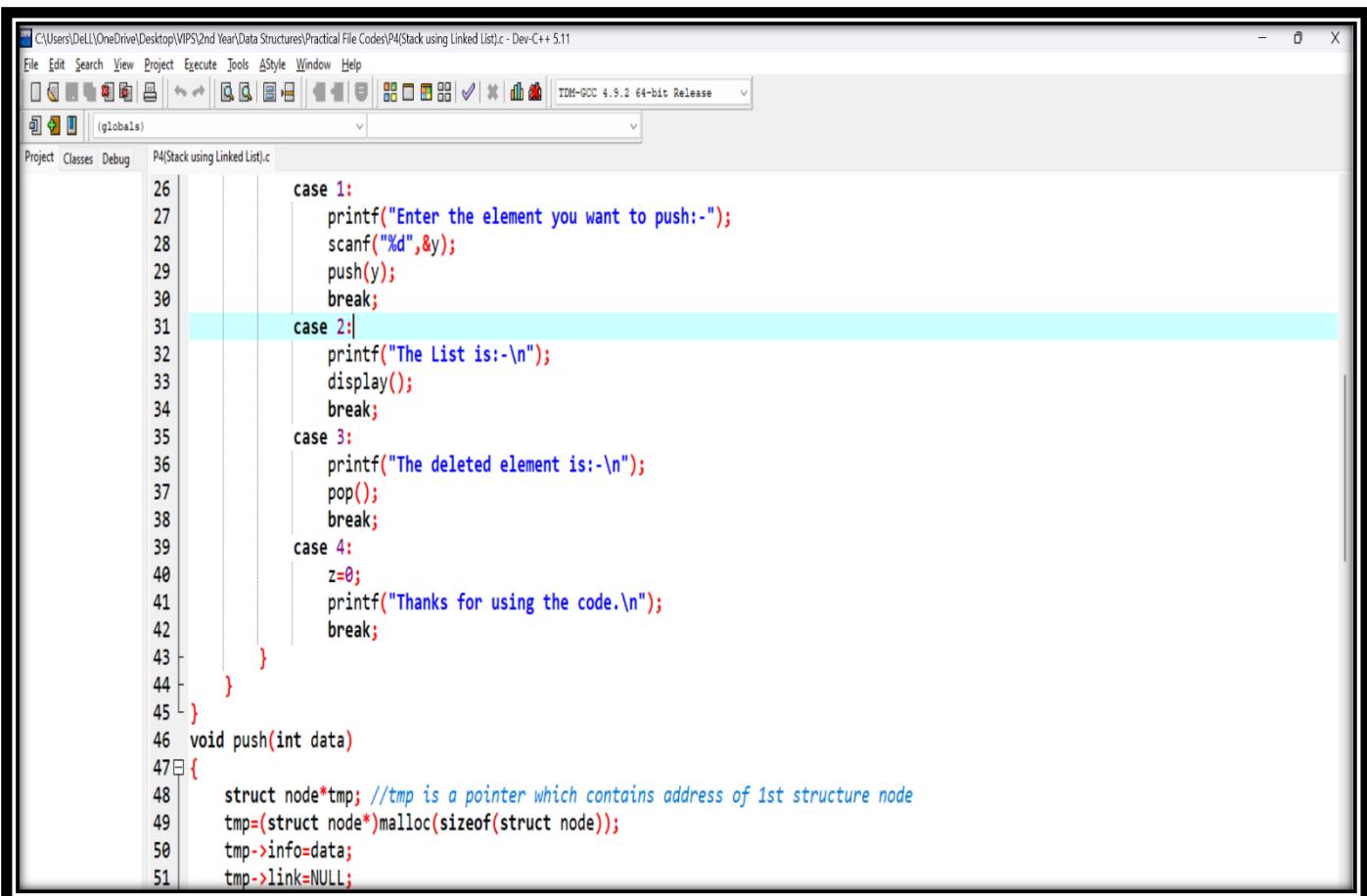
The screenshot shows the Dev-C++ IDE interface with the following details:

- Title Bar:** C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P4(Stack using Linked List).c - Dev-C++ 5.11
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help
- Toolbar:** Standard Dev-C++ toolbar with icons for file operations.
- Project Explorer:** Shows the project name P4(Stack using Linked List).c
- Code Editor:**

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 struct node
4 {
5     int info;
6     struct node*link;
7 }*top;
8 void push(int data);
9 void display();
10 void pop();
11 void main()
12 {
13     int x, y, z=1;
14     printf("-----Stack using Linked List-----\n");
15     while(z==1)
16     {
17         printf("Select the operation you want to perform:-\n");
18         printf("1.Push\n");
19         printf("2.Display\n");
20         printf("3.Pop\n");
21         printf("4.Exit\n");
22         printf("Enter your choice:-");
23         scanf("%d",&x);
24         switch(x)
25         {
26             case 1:

```
- Compiler Output:**
 - Errors: 0
 - Warnings: 0
 - Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P4(Stack using Linked List).exe
 - Output Size: 129.870171875 KiB
 - Compilation Time: 0.22s
- Status Bar:** Line: 31, Col: 20, Sel: 0, Lines: 94, Length: 1505, Insert, Done parsing in 0.094 seconds



The screenshot shows the Dev-C++ IDE interface with the following details:

- Title Bar:** C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P4(Stack using Linked List).c - Dev-C++ 5.11
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help
- Toolbar:** Standard Dev-C++ toolbar with icons for file operations.
- Project Explorer:** Shows the project name P4(Stack using Linked List).c
- Code Editor:**

```

26     case 1:
27         printf("Enter the element you want to push:-");
28         scanf("%d",&y);
29         push(y);
30         break;
31     case 2:
32         printf("The List is:-\n");
33         display();
34         break;
35     case 3:
36         printf("The deleted element is:-\n");
37         pop();
38         break;
39     case 4:
40         z=0;
41         printf("Thanks for using the code.\n");
42         break;
43     }
44 }
45 }
46 void push(int data)
47 {
48     struct node*tmp; //tmp is a pointer which contains address of 1st structure node
49     tmp=(struct node*)malloc(sizeof(struct node));
50     tmp->info=data;
51     tmp->link=NULL;

```

KUNSH SABHARWAL

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P4(Stack using Linked List).c - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project Classes Debug P4(Stack using Linked List).c

```
52     if(top==NULL)
53     {
54         top=tmp;
55     }
56     else
57     {
58         tmp->link=top;
59         top=tmp;
60     }
61 }
62 void display()
63 {
64     struct node*tmp;
65     tmp=top;
66     if(tmp==NULL)
67     {
68         printf("No Element in List.\n");
69     }
70     else
71     {
72         while(tmp!=NULL)
73         {
74             printf("%d\n",tmp->info);
75             tmp=tmp->link;
76         }
77     }
78 }
```

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P4(Stack using Linked List).c - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project Classes Debug P4(Stack using Linked List).c

```
69 }
70 else
71 {
72     while(tmp!=NULL)
73     {
74         printf("%d\n",tmp->info);
75         tmp=tmp->link;
76     }
77 }
78 }
79 void pop()
80 {
81     struct node*tmp;
82     if(top==NULL)
83     {
84         printf("No element in list.\n");
85     }
86     else
87     {
88         tmp=top;
89         printf("%d\n",tmp->info);
90         top=top->link;
91         free(tmp);
92     }
93 }
94 }
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

```
=====
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P4(Stack using Linked List).exe
- Output Size: 129.8701171875 KB
- Compilation Time: 0.22s
```

Line: 31 Col: 20 Sel: 0 Lines: 94 Length: 1505 Insert Done parsing in 0.094 seconds

Outputs:

```

C:\Users\DeLL\OneDrive\Desktop
-----Stack using Linked List-----
Select the operation you want to perform:-
1.Push
2.Display
3.Pop
4.Exit
Enter your choice:-1
Enter the element you want to push:-5
Select the operation you want to perform:-
1.Push
2.Display
3.Pop
4.Exit
Enter your choice:-1
Enter the element you want to push:-10
Select the operation you want to perform:-
1.Push
2.Display
3.Pop
4.Exit
Enter your choice:-1
Enter the element you want to push:-15
Select the operation you want to perform:-
1.Push
2.Display
3.Pop
4.Exit
Enter your choice:-2
The List is:-
15
10
5
Select the operation you want to perform:-
1.Push
2.Display
3.Pop
4.Exit
Enter your choice:-3
The deleted element is:-
15
Select the operation you want to perform:-
1.Push
2.Display
3.Pop
4.Exit
Enter your choice:-2

```

```

C:\Users\DeLL\OneDrive\Desktop
The List is:-
10
5
Select the operation you want to perform:-
1.Push
2.Display
3.Pop
4.Exit
Enter your choice:-3
The deleted element is:-
10
Select the operation you want to perform:-
1.Push
2.Display
3.Pop
4.Exit
Enter your choice:-3
The deleted element is:-
5
Select the operation you want to perform:-
1.Push
2.Display
3.Pop
4.Exit
Enter your choice:-3
The deleted element is:-
No element in list.
Select the operation you want to perform:-
1.Push
2.Display
3.Pop
4.Exit
Enter your choice:-2
The List is:-
No Element in List.
Select the operation you want to perform:-
1.Push
2.Display
3.Pop
4.Exit
Enter your choice:-4
Thanks for using the code.

-----
Process exited after 24.85 seconds with return value 0
Press any key to continue . . . |

```

Learning Outcome:

EXPERIMENT 5

Problem statement: Write a program to implement a Linked List using Queue and perform operations such as push, pop and display the list.

Algorithm:

Source Code:

C:\Users\DeLL\OneDrive\Desktop\VIP'S\2nd Year\Data Structures\Practical File Codes\P5(Queue using Linked List).c - [Executing] - Dev-C++ 5.11

```

File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug P5(Queue using Linked List).c
1 #include<stdio.h>
2 #include<stdlib.h>
3 struct que
4 {
5     int data;
6     struct que*link;
7 }*f,*r=NULL;
8 void enqueue(int x);
9 void display();
10 void dequeue();
11 void main()
12 {
13     int x, y, z=1;
14     printf("-----Queue using Linked List-----\n");
15     while(z==1)
16     {
17         printf("Select the operation you want to perform:-\n");
18         printf("1.Enqueue\n");
19         printf("2.Display\n");
20         printf("3.Dequeue\n");
21         printf("4.Exit\n");
22         printf("Enter your choice:-");
23         scanf("%d",&x);
24         switch(x)
25         {
26             case 1:

```

Compiler Resources Compile Log Debug Find Results Close

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIP'S\2nd Year\Data Structures\Practical File Codes\P5(Queue using Linked List).exe
- Output Size: 129.887653125 Kib
- Compilation Time: 0.20s

Shorten compiler paths

Line: 70 Col: 38 Sel: 0 Lines: 95 Length: 1483 Insert Done parsing in 0 seconds

C:\Users\DeLL\OneDrive\Desktop\VIP'S\2nd Year\Data Structures\Practical File Codes\P5(Queue using Linked List).c - [Executing] - Dev-C++ 5.11

```

File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug P5(Queue using Linked List).c
27     printf("Enter the element you want to insert:-");
28     scanf("%d",&y);
29     enqueue(y);
30     break;
31     case 2:
32         printf("The List is:-\n");
33         display();
34         break;
35     case 3:
36         printf("The Deleted element is:-\n");
37         dequeue();
38         break;
39     case 4:
40         z=0;
41         printf("Thanks for using the code.\n");
42         break;
43     default:
44         printf("Invalid Choice!\n");
45     }
46 }
47 }
48 void enqueue(int x)
49 {
50     struct que*tmp;
51     tmp=(struct que*)malloc(sizeof(struct que));
52     tmp->data=x;

```

KUNSH SABHARWAL

C:\Users\DeLL\OneDrive\Desktop\VIIPS2nd Year\Data Structures\Practical File Codes\PS\Queue using Linked List.c - [Executing] - Dev-C++ 5.11

```
File Edit Search View Project Execute Tools AStyle Window Help
globals Project Classes Debug P5\Queue using Linked List.c

53     tmp->link=NULL;
54     if(f==NULL && r==NULL)
55     {
56         f=tmp;
57         r=tmp;
58     }
59     else
60     {
61         r->link=tmp;
62         r=tmp;
63     }
64 }
65 void display()
66 {
67     struct que*tmp;
68     if(f==NULL)
69     {
70         printf("No Element in list.\n");
71     }
72     else
73     {
74         tmp=f;
75         while(tmp!=NULL)
76         {
77             printf("%d\n",tmp->data);
78             tmp=tmp->link;
79         }
80     }
81 }
82 void dequeue()
83 {
84     struct que*tmp;
85     if(f==NULL)
86     {
87         printf("No Element in List.\n");
88     }
89     else
90     {
91         printf("%d\n",f->data);
92         f=f->link;
93     }
94 }
```

C:\Users\DeLL\OneDrive\Desktop\VIIPS2nd Year\Data Structures\Practical File Codes\PS\Queue using Linked List.c - [Executing] - Dev-C++ 5.11

```
File Edit Search View Project Execute Tools AStyle Window Help
globals Project Classes Debug P5\Queue using Linked List.c

70     printf("No Element in list.\n");
71 }
72 else
73 {
74     tmp=f;
75     while(tmp!=NULL)
76     {
77         printf("%d\n",tmp->data);
78         tmp=tmp->link;
79     }
80 }
81 }
82 void dequeue()
83 {
84     struct que*tmp;
85     if(f==NULL)
86     {
87         printf("No Element in List.\n");
88     }
89     else
90     {
91         printf("%d\n",f->data);
92         f=f->link;
93     }
94 }
```

Outputs:

```
-----Queue using Linked List-----
Select the operation you want to perform:-
1.Enqueue
2.Display
3.Dequeue
4.Exit
Enter your choice:-1
Enter the element you want to insert:-5
Select the operation you want to perform:-
1.Enqueue
2.Display
3.Dequeue
4.Exit
Enter your choice:-1
Enter the element you want to insert:-10
Select the operation you want to perform:-
1.Enqueue
2.Display
3.Dequeue
4.Exit
Enter your choice:-1
Enter the element you want to insert:-15
Select the operation you want to perform:-
1.Enqueue
2.Display
3.Dequeue
4.Exit
Enter your choice:-2
The List is:-
5
10
15
Select the operation you want to perform:-
1.Enqueue
2.Display
3.Dequeue
4.Exit
Enter your choice:-3
The Deleted element is:-
5
Select the operation you want to perform:-
1.Enqueue
2.Display
3.Dequeue
4.Exit
Enter your choice:-2
The List is:-
No Element in list.
Select the operation you want to perform:-
1.Enqueue
2.Display
3.Dequeue
4.Exit
Enter your choice:-4
Thanks for using the code.
```

```
The List is:-
10
15
Select the operation you want to perform:-
1.Enqueue
2.Display
3.Dequeue
4.Exit
Enter your choice:-3
The Deleted element is:-
10
Select the operation you want to perform:-
1.Enqueue
2.Display
3.Dequeue
4.Exit
Enter your choice:-2
The List is:-
15
Select the operation you want to perform:-
1.Enqueue
2.Display
3.Dequeue
4.Exit
Enter your choice:-3
The Deleted element is:-
15
Select the operation you want to perform:-
1.Enqueue
2.Display
3.Dequeue
4.Exit
Enter your choice:-2
The List is:-
No Element in list.
Select the operation you want to perform:-
1.Enqueue
2.Display
3.Dequeue
4.Exit
Enter your choice:-4
Thanks for using the code.

-----
Process exited after 23.62 seconds with return value 0
Press any key to continue . . .
```

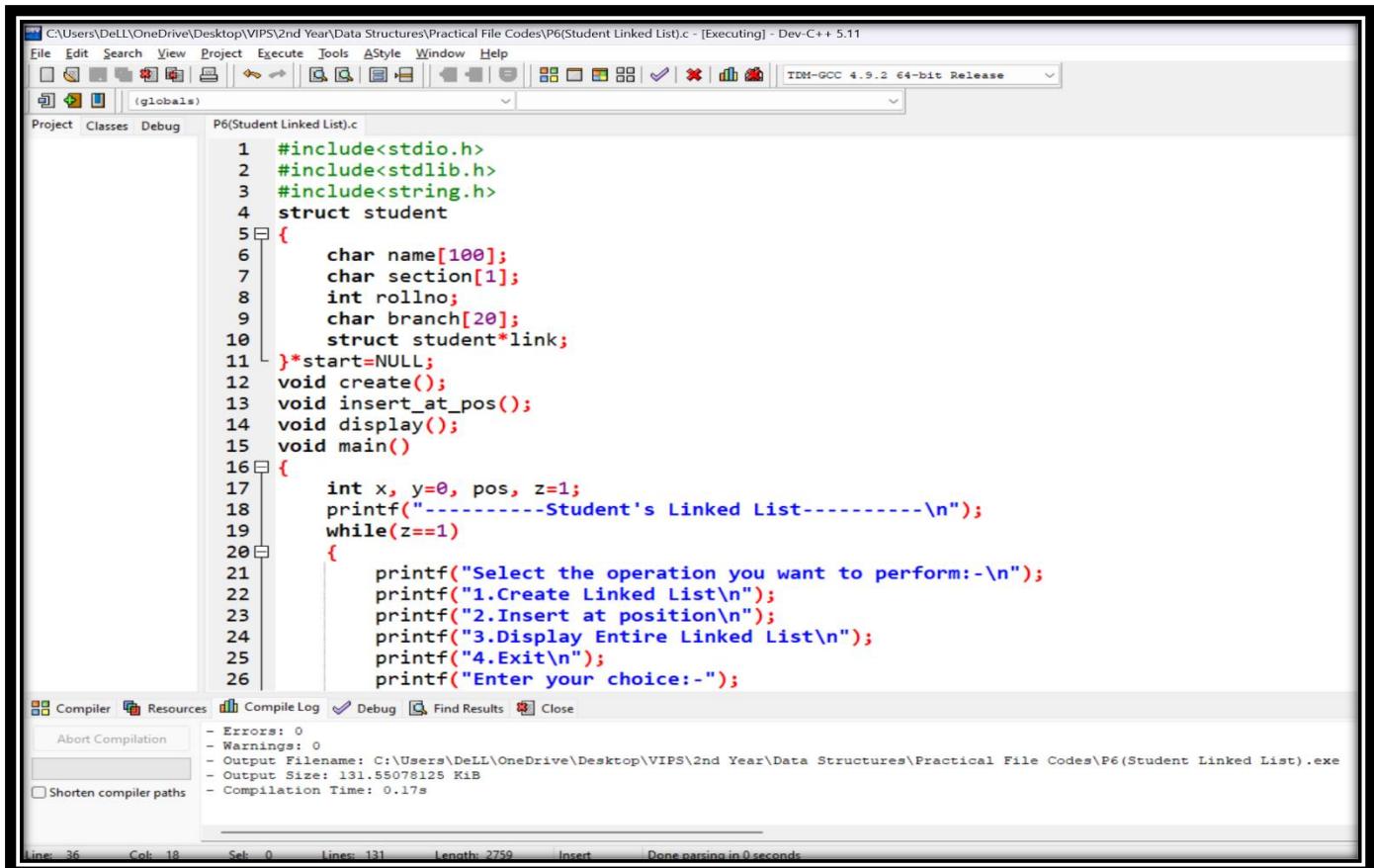
Learning Outcome:

EXPERIMENT 6

Problem statement: Write a program to create a linked list with nodes having information about student's name, branch, section and roll number. Also insert a new node at a position specified by the user.

Algorithm:

Source Code:



The screenshot shows the Dev-C++ IDE interface with the following details:

- Title Bar:** C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P6(Student Linked List).c - [Executing] - Dev-C++ 5.11
- Toolbar:** Standard Dev-C++ toolbar with icons for file operations, project management, and compilation.
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help.
- Project Explorer:** Shows the project "P6(Student Linked List).c" with files listed.
- Code Editor:**

```

1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 struct student
5 {
6     char name[100];
7     char section[1];
8     int rollno;
9     char branch[20];
10    struct student*link;
11 }*start=NULL;
12 void create();
13 void insert_at_pos();
14 void display();
15 void main()
16 {
17     int x, y=0, pos, z=1;
18     printf("-----Student's Linked List-----\n");
19     while(z==1)
20     {
21         printf("Select the operation you want to perform:-\n");
22         printf("1.Create Linked List\n");
23         printf("2.Insert at position\n");
24         printf("3.Display Entire Linked List\n");
25         printf("4.Exit\n");
26         printf("Enter your choice:-");

```
- Compiler Tab:** Shows compilation results:
 - Errors: 0
 - Warnings: 0
 - Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P6(Student Linked List).exe
 - Output Size: 131.55078125 KiB
 - Compilation Time: 0.17s
- Status Bar:** Line: 36, Col: 18, Sel: 0, Lines: 131, Length: 2759, Insert, Done parsing in 0 seconds.

KUNSH SABHARWAL

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P6(Student Linked List).c - [Executing] - Dev-C++ 5.11

```
27
28
29
30     scanf("%d",&x);
31     switch(x)
32     {
33         case 1:
34             if (y==0)
35             {
36                 printf("Creating New Linked List:-\n");
37                 create();
38                 y=1;
39             }
39         else
40             {
41                 printf("Linked List already exists! Try inserting at position.\n");
42             }
43             break;
44         case 2:
45             insert_at_pos();
46             break;
47         case 3:
48             display();
49             break;
50         case 4:
51             z=0;
52             printf("Thanks for using the code.\n");
53             break;
54         default:
55     }
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P6(Student Linked List).exe
- Output Size: 131.55078125 Kib
- Compilation Time: 0.17s

Line: 36 Col: 18 Sel: 0 Lines: 131 Length: 2759 Insert Done parsing in 0 seconds

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P6(Student Linked List).c - [Executing] - Dev-C++ 5.11

```
53         printf("Invalid Choice!\n");
54     }
55 }
56 }
57 void create()
58 {
59     struct student*q, *tmp;
60     char name[100], sec[1], branch[20];
61     int rollno;
62     tmp=(struct student*)malloc(sizeof(struct student));
63     printf("Enter the students name:-");
64     scanf("%s",&name);
65     strcpy(tmp->name,name);
66     printf("Enter the students branch:-");
67     scanf("%s",&branch);
68     strcpy(tmp->branch,branch);
69     printf("Enter the students section:-");
70     scanf("%s",&sec);
71     strcpy(tmp->section,sec);
72     printf("Enter the students roll number:-");
73     scanf("%d",&rollno);
74     tmp->rollno=rollno;
75     tmp->link=NULL;
76     start=tmp;
77 }
78 void insert_at_pos()
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P6(Student Linked List).exe
- Output Size: 131.55078125 Kib
- Compilation Time: 0.17s

Line: 36 Col: 18 Sel: 0 Lines: 131 Length: 2759 Insert Done parsing in 0 seconds

KUNSH SABHARWAL

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P6(Student Linked List).c - [Executing] - Dev-C++ 5.11

```
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug P6(Student Linked List).c
79 {
80     struct student*tmp, *q;
81     char name[100], sec[1], branch[20];
82     int rollno;
83     int pos;
84     int i;
85     q=start;
86     printf("Enter the position you want to insert at:-");
87     scanf("%d",&pos);
88     for(i=0;i<pos-1;i++)
89     {
90         q=q->link;
91         if(q==NULL)
92         {
93             printf("Invalid Position!\n");
94             return;
95         }
96         tmp=(struct student*)malloc(sizeof(struct student));
97         printf("Enter the students name:-");
98         scanf("%s",&name);
99         strcpy(tmp->name,name);
100        printf("Enter the students branch:-");
101        scanf("%s",&branch);
102        strcpy(tmp->branch,branch);
103        printf("Enter the students section:-");
104    }
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P6(Student Linked List).exe
- Output Size: 131.55078125 KiB
- Compilation Time: 0.17s

Line: 36 Col: 18 Sel: 0 Lines: 131 Length: 2759 Insert Done parsing in 0 seconds

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P6(Student Linked List).c - [Executing] - Dev-C++ 5.11

```
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug P6(Student Linked List).c
104     printf("Enter the students section:-");
105     scanf("%s",&sec);
106     strcpy(tmp->section,sec);
107     printf("Enter the students roll number:-");
108     scanf("%d",&rollno);
109     tmp->rollno=rollno;
110     tmp->link=q->link;
111     q->link=tmp;
112 }
113 void display()
114 {
115     struct student*q;
116     if(start==NULL)
117     {
118         printf("List is empty!\n");
119         return;
120     }
121     q=start;
122     printf("Linked List of students is:-\n");
123     while(q!=NULL)
124     {
125         printf("Name:-%s\n",q->name);
126         printf("Branch:-%s\n",q->branch);
127         printf("Section:-%s\n",q->section);
128         printf("Roll No:-%d\n",q->rollno);
129         q=q->link;
130     }
131 }
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P6(Student Linked List).exe

Line: 36 Col: 18 Sel: 0 Lines: 131 Length: 2759 Insert Done parsing in 0 seconds

Output:

```
C:\Users\DeLL\OneDrive\Desktop + ▾
-----Student's Linked List-----
Select the operation you want to perform:-
1.Create Linked List
2.Insert at position
3.Display Entire Linked List
4.Exit
Enter your choice:-1
Creating New Linked List:-
Enter the students name:-Kunsh
Enter the students branch:-AIML
Enter the students section:-A
Enter the students roll number:-1
Select the operation you want to perform:-
1.Create Linked List
2.Insert at position
3.Display Entire Linked List
4.Exit
Enter your choice:-1
Linked List already exists! Try inserting at position.
Select the operation you want to perform:-
1.Create Linked List
2.Insert at position
3.Display Entire Linked List
4.Exit
Enter your choice:-3
Linked List of students is:-
Name:-Kunsh
Branch:-AIML
Section:-A
Roll No:-1
Select the operation you want to perform:-
1.Create Linked List
2.Insert at position
3.Display Entire Linked List
4.Exit
Enter your choice:-2
Enter the position you want to insert at:-1
Enter the students name:-Raj
Enter the students branch:-AIDS
Enter the students section:-A
Enter the students roll number:-2
Select the operation you want to perform:-
1.Create Linked List
```

```
C:\Users\DeLL\OneDrive\Desktop> +
1.Create Linked List
2.Insert at position
3.Display Entire Linked List
4.Exit
Enter your choice:-3
Linked List of students is:-
Name:-Kunsh
Branch:-AIML
Section:-A
Roll No:-1
Name:-Raj
Branch:-AIDS
Section:-A
Roll No:-2
Select the operation you want to perform:-
1.Create Linked List
2.Insert at position
3.Display Entire Linked List
4.Exit
Enter your choice:-2
Enter the position you want to insert at:-2
Enter the students name:-Ram
Enter the students branch:-AIML
Enter the students section:-B
Enter the students roll number:-3
Select the operation you want to perform:-
1.Create Linked List
2.Insert at position
3.Display Entire Linked List
4.Exit
Enter your choice:-3
Linked List of students is:-
Name:-Kunsh
Branch:-AIML
Section:-A
Roll No:-1
Name:-Raj
Branch:-AIDS
Section:-A
Roll No:-2
Name:-Ram
Branch:-AIML
Section:-B
Roll No:-3
Select the operation you want to perform:-
1.Create Linked List
2.Insert at position
3.Display Entire Linked List
4.Exit
```

```
C:\Users\DeLL\OneDrive\Desktop> +
4.Exit
Enter your choice:-2
Enter the position you want to insert at:-1
Enter the students name:-Rahul
Enter the students branch:-IIOT
Enter the students section:-A
Enter the students roll number:-4
Select the operation you want to perform:-
1.Create Linked List
2.Insert at position
3.Display Entire Linked List
4.Exit
Enter your choice:-3
Linked List of students is:-
Name:-Kunsh
Branch:-AIML
Section:-A
Roll No:-1
Name:-Rahul
Branch:-IIOT
Section:-A
Roll No:-4
Name:-Raj
Branch:-AIDS
Section:-A
Roll No:-2
Name:-Ram
Branch:-AIML
Section:-B
Roll No:-3
Select the operation you want to perform:-
1.Create Linked List
2.Insert at position
3.Display Entire Linked List
4.Exit
Enter your choice:-2
Enter the position you want to insert at:-7
Invalid Position!
Select the operation you want to perform:-
1.Create Linked List
2.Insert at position
3.Display Entire Linked List
4.Exit
Enter your choice:-4
Thanks for using the code.

-----
Process exited after 125.9 seconds with return value 0
Press any key to continue . . . |
```

Learning Outcome:

EXPERIMENT 7

Problem statement: Write a program to create a doubly linked list with nodes having information about an employee and perform insertion at front of doubly linked list and perform deletion at the end of that doubly linked list

Algorithm:

Source Code:

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P7(Employee DLL).c - Dev-C++ 5.11

```

File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug P7(Employee DLL).c
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 struct employee
5 {
6     char name[100];
7     int empid;
8     char department[50];
9     char designation[50];
10    int salary;
11    struct employee*prev;
12    struct employee*next;
13 }*start;
14 void create();
15 void insert_at_beg();
16 void del_at_end();
17 void display();
18 void main()
19 {
20     int x, y=0, pos, z=1;
21     printf("-----Employee Linked List-----\n");
22     while(z==1)
23     {
24         printf("Select the operation you want to perform:-\n");
25         printf("1. Create a Linked List\n");
26         printf("2. Insert at beginning\n");

```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P7(Employee DLL).exe
- Output Size: 131.0537109375 KiB
- Compilation Time: 0.16s

Line: 54 Col: 27 Sel: 0 Lines: 158 Length: 3350 Insert Done parsing in 0.016 seconds

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P7(Employee DLL).c - Dev-C++ 5.11

```

File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug P7(Employee DLL).c
26     printf("2. Insert at beginning\n");
27     printf("3. Deletion at end\n");
28     printf("4. Display\n");
29     printf("5. Exit\n");
30     printf("Enter your choice:-");
31     scanf("%d",&x);
32     switch(x)
33     {
34         case 1:
35             if(y==0)
36             {
37                 printf("Creating a new Linked List:-\n");
38                 create();
39                 y=1;
40             }
41             else
42             {
43                 printf("Linked List already exists! Try inserting at beginning.\n");
44             }
45             break;
46         case 2:
47             insert_at_beg();
48             break;
49         case 3:
50             del_at_end();
51             printf("Last Element of Linked List deleted successfully!\n");
52             break;
53         case 4:
54             display();
55             break;
56     }

```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Line: 53 Col: 20 Sel: 0 Lines: 158 Length: 3353 Insert Done parsing in 0.015 seconds

KUNSH SABHARWAL

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P7(Employee DLL).c - Dev-C++ 5.11

```
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug P7(Employee DLL).c
53     case 4:
54         display();
55         break;
56     case 5:
57         z=0;
58         printf("Thanks for using the code!\n");
59         break;
60     default:
61         printf("Invalid Choice!\n");
62     }
63 }
64 void create()
65 {
66     struct employee*tmp,*q;
67     char name[100], department[50], designation[50];
68     int empid, salary;
69     tmp=(struct employee*)malloc(sizeof(struct employee));
70     printf("Enter the Employee's Name:-");
71     scanf("%s",&name);
72     strcpy(tmp->name,name);
73     printf("Enter the Employee's ID:-");
74     scanf("%d",&empid);
75     tmp->empid=empid;
76     printf("Enter the Employee's Department:-");
77     scanf("%s",&department);
78 }
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P7(Employee DLL).exe
- Output Size: 131.0537109375 KiB
- Compilation Time: 0.16s

Line: 54 Col: 27 Sel: 0 Lines: 158 Length: 3350 Insert Done parsing in 0.016 seconds

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P7(Employee DLL).c - Dev-C++ 5.11

```
File Edit Search View Project Execute Tools AStyle Window Help
Project Classes Debug P7(Employee DLL).c
79     strcpy(tmp->department,department);
80     printf("Enter the Employee's Designation:-");
81     scanf("%s",&designation);
82     strcpy(tmp->designation,designation);
83     printf("Enter the Employee's Salary:-");
84     scanf("%d",&salary);
85     tmp->salary=salary;
86     tmp->prev=NULL;
87     tmp->next=NULL;
88     start=tmp;
89 }
90 void insert_at_beg()
91 {
92     struct employee*tmp;
93     char name[100], department[50], designation[50];
94     int empid, salary;
95     tmp=(struct employee*)malloc(sizeof(struct employee));
96     printf("Enter the Employee's Name:-");
97     scanf("%s",&name);
98     strcpy(tmp->name,name);
99     printf("Enter the Employee's ID:-");
100    scanf("%d",&empid);
101    tmp->empid=empid;
102    printf("Enter the Employee's Department:-");
103    scanf("%s",&department);
104    strcpy(tmp->department,department);
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P7(Employee DLL).exe
- Output Size: 131.0537109375 KiB
- Compilation Time: 0.16s

Line: 54 Col: 27 Sel: 0 Lines: 158 Length: 3350 Insert Done parsing in 0.016 seconds

KUNSH SABHARWAL

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P7(Employee DLL).c - Dev-C++ 5.11

```
105     printf("Enter the Employee's Designation:-");
106     scanf("%s",&designation);
107     strcpy(tmp->designation,designation);
108     printf("Enter the Employee's Salary:-");
109     scanf("%d",&salary);
110     tmp->salary=salary;
111     tmp->prev=NULL;
112     tmp->next=NULL;
113     tmp->next=start;
114     start->prev=tmp;
115     start=tmp;
116 }
117 void del_at_end()
118 {
119     struct employee*tmp=start;
120     if(start==NULL)
121     {
122         printf("List is empty!");
123         return;
124     }
125     else if(tmp->next==NULL)
126     {
127         start=NULL;
128         return;
129     }
130     while(tmp->next!=NULL)
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P7(Employee DLL).exe
- Output Size: 131.0537109375 KiB
- Compilation Time: 0.16s

Line: 54 Col: 27 Sel: 0 Lines: 158 Length: 3350 Insert Done parsing in 0.016 seconds

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P7(Employee DLL).c - Dev-C++ 5.11

```
129 }
130 while(tmp->next!=NULL)
131 {
132     tmp=tmp->next;
133 }
134 struct employee*q=tmp->prev;
135 q->next=NULL;
136 free(tmp);
137 }
138 void display()
139 {
140     struct employee*q;
141     if(start==NULL)
142     {
143         printf("List is empty!");
144     }
145     else
146     {
147         q=start;
148         while(q!=NULL)
149     {
150         printf("%s\n",q->name);
151         printf("%d\n",q->empid);
152         printf("%s\n",q->department);
153         printf("%s\n",q->designation);
154         printf("%d\n",q->salary);
155         q=q->next;
156     }
157 }
```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

Line: 54 Col: 27 Sel: 0 Lines: 158 Length: 3350 Insert Done parsing in 0.016 seconds

Output:

```
C:\Users\DeLL\OneDrive\Desktop> -----
-----Employee Linked List-----
Select the operation you want to perform:-
1. Create a Linked List
2. Insert at beginning
3. Deletion at end
4. Display
5. Exit
Enter your choice:-1
Creating a new Linked List:-
Enter the Employee's Name:-Kunsh
Enter the Employee's ID:-1
Enter the Employee's Department:-Sales
Enter the Employee's Designation:-VP
Enter the Employee's Salary:-100000
Select the operation you want to perform:-
1. Create a Linked List
2. Insert at beginning
3. Deletion at end
4. Display
5. Exit
Enter your choice:-1
Linked List already exists! Try inserting at beginning.
Select the operation you want to perform:-
1. Create a Linked List
2. Insert at beginning
3. Deletion at end
4. Display
5. Exit
Enter your choice:-4
Kunsh
1
Sales
VP
100000
Select the operation you want to perform:-
1. Create a Linked List
2. Insert at beginning
3. Deletion at end
4. Display
5. Exit
Enter your choice:-2
Enter the Employee's Name:-Ram
Enter the Employee's ID:-2
Enter the Employee's Department:-HR
Enter the Employee's Designation:-Manager
Enter the Employee's Salary:-50000
```

```
C:\Users\DeLL\OneDrive\Desktop> -----
Select the operation you want to perform:-
1. Create a Linked List
2. Insert at beginning
3. Deletion at end
4. Display
5. Exit
Enter your choice:-4
Ram
2
HR
Manager
50000
Kunsh
1
Sales
VP
100000
Select the operation you want to perform:-
1. Create a Linked List
2. Insert at beginning
3. Deletion at end
4. Display
5. Exit
Enter your choice:-3
Last Element of Linked List deleted successfully!
Select the operation you want to perform:-
1. Create a Linked List
2. Insert at beginning
3. Deletion at end
4. Display
5. Exit
Enter your choice:-4
Kunsh
1
Sales
VP
100000
Select the operation you want to perform:-
1. Create a Linked List
2. Insert at beginning
3. Deletion at end
4. Display
5. Exit
Enter your choice:-8
Invalid Choice!
Select the operation you want to perform:-
1. Create a Linked List
2. Insert at beginning
3. Deletion at end
4. Display
5. Exit
Enter your choice:-5
Thanks for using the code!
```

Learning Outcome:

EXPERIMENT 8

Problem statement: Write a program to create a Circular Linked List having information about a college and perform insertion at the front end and perform deletion at the end.

Algorithm:

Source Code:

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P8 (CLL).c - [Executing] - Dev-C++ 5.11

```

File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release

Project Classes Debug P8 (CLL).c
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<string.h>
4 struct college
5 {
6     int clgid;
7     char clgname[100];
8     int stdcount;
9     struct college*next;
10 }*start=NULL;
11 void create();
12 void insert_at_beg();
13 void del_at_end();
14 void display();
15 void main()
16 {
17     int x, y=0, z=1;
18     printf("-----College Linked List-----\n");
19     while(z==1)
20     {
21         printf("Select the operation you want to perform:-\n");
22         printf("1. Create a Linked List\n");
23         printf("2. Insertion at Beginning\n");
24         printf("3. Deletion at End\n");
25         printf("4. Display\n");

```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P8 (CLL).exe
- Output Size: 131.0537109375 KiB
- Compilation Time: 0.16s

Line: 1 Col: 1 Sel: 0 Lines: 146 Length: 2792 Insert Done parsing in 0.078 seconds

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P8 (CLL).c - [Executing] - Dev-C++ 5.11

```

File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release

Project Classes Debug P8 (CLL).c
26     printf("5. Exit\n");
27     printf("Enter your choice:-");
28     scanf("%d",&x);
29     switch(x)
30     {
31         case 1:
32             if(y==0)
33             {
34                 printf("Creating a New Linked List\n");
35                 create();
36                 y=1;
37             }
38             else
39             {
40                 printf("Linked List already exists! Try inserting at beginning.\n");
41             }
42             break;
43         case 2:
44             insert_at_beg();
45             break;
46         case 3:
47             del_at_end();
48             printf("Last Element deleted successfully!\n");
49             break;
50         case 4:

```

Compiler Resources Compile Log Debug Find Results Close

Abort Compilation

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P8 (CLL).exe
- Output Size: 131.0537109375 KiB
- Compilation Time: 0.16s

Line: 1 Col: 1 Sel: 0 Lines: 146 Length: 2792 Insert Done parsing in 0.078 seconds

KUNSH SABHARWAL

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P8 (CLL).c - [Executing] - Dev-C++ 5.11

```
File Edit Search View Project Execute Tools AStyle Window Help
(globals) TDM-GCC 4.9.2 64-bit Release

Project Classes Debug P8 (CLL).c
51     display();
52     break;
53     case 5:
54         z=0;
55         printf("Thanks for using this code!\n");
56     default:
57         printf("Invalid Choice!");
58     }
59 }
60 }
61 void create()
62 {
63     struct college*tmp;
64     int clgid, stdcount;
65     char clgname[100];
66     tmp=(struct college*)malloc(sizeof(struct college));
67     printf("Enter the College ID:-");
68     scanf("%d",&clgid);
69     tmp->clgid=clgid;
70     printf("Enter the College Name:-");
71     scanf("%s",&clgname);
72     strcpy(tmp->clgname,clgname);
73     printf("Enter Number of Students in College:-");
74     scanf("%d",&stdcount);
75     tmp->stdcount=stdcount;

Compiler Resources Compile Log Debug Find Results Close
----- Errors: 0
----- Warnings: 0
----- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P8 (CLL).exe
----- Output Size: 131.0537109375 KiB
----- Compilation Time: 0.16s

Line: 1 Col: 1 Sel: 0 Lines: 146 Length: 2792 Insert Done parsing in 0.078 seconds
```

C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P8 (CLL).c - [Executing] - Dev-C++ 5.11

```
File Edit Search View Project Execute Tools AStyle Window Help
(globals) TDM-GCC 4.9.2 64-bit Release

Project Classes Debug P8 (CLL).c
76     start=tmp;
77     tmp->next=start;
78 }
79 void insert_at_beg()
80 {
81     struct college*tmp;
82     int clgid, stdcount;
83     char clgname[100];
84     tmp=(struct college*)malloc(sizeof(struct college));
85     printf("Enter the College ID:-");
86     scanf("%d",&clgid);
87     tmp->clgid=clgid;
88     printf("Enter the College Name:-");
89     scanf("%s",&clgname);
90     strcpy(tmp->clgname,clgname);
91     printf("Enter Number of Students in College:-");
92     scanf("%d",&stdcount);
93     tmp->stdcount=stdcount;
94     tmp->next=NULL;
95     struct college*q=start;
96     while(q->next!=start)
97     {
98         q=q->next;
99     }
100    tmp->next=start;

Compiler Resources Compile Log Debug Find Results Close
----- Errors: 0
----- Warnings: 0
----- Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P8 (CLL).exe
----- Output Size: 131.0537109375 KiB
----- Compilation Time: 0.16s

Line: 1 Col: 1 Sel: 0 Lines: 146 Length: 2792 Insert Done parsing in 0.078 seconds
```

KUNSH SABHARWAL

The screenshot shows the Dev-C++ IDE interface with the following details:

- Title Bar:** C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P8 (CLL).c - [Executing] - Dev-C++ 5.11
- Toolbar:** Standard Dev-C++ toolbar with icons for file operations, project management, and tools.
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help.
- Toolbox:** Global toolbox with icons for file operations, project management, and tools.
- Compiler Status:** TDM-GCC 4.9.2 64-bit Release
- Code Editor:** The code for P8 (CLL).c is displayed, showing functions for insertion at the beginning, deletion at the end, and displaying the list.
- Compiler Log:** Shows compilation results:
 - Errors: 0
 - Warnings: 0
 - Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P8 (CLL).exe
 - Output Size: 131.0537109375 Kib
 - Compilation Time: 0.16s
- Status Bar:** Line: 1 Col: 1 Sel: 0 Lines: 146 Length: 2792 Insert Done parsing in 0.078 seconds

The screenshot shows the Dev-C++ IDE interface with the following details:

- Title Bar:** C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P8 (CLL).c - [Executing] - Dev-C++ 5.11
- Toolbar:** Standard Dev-C++ toolbar with icons for file operations, project management, and tools.
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help.
- Toolbox:** Global toolbox with icons for file operations, project management, and tools.
- Compiler Status:** TDM-GCC 4.9.2 64-bit Release
- Code Editor:** The code for P8 (CLL).c is displayed, showing the display function which prints the linked list elements.
- Compiler Log:** Shows compilation results:
 - Errors: 0
 - Warnings: 0
 - Output Filename: C:\Users\DeLL\OneDrive\Desktop\VIPS\2nd Year\Data Structures\Practical File Codes\P8 (CLL).exe
 - Output Size: 131.0537109375 Kib
 - Compilation Time: 0.16s
- Status Bar:** Line: 1 Col: 1 Sel: 0 Lines: 146 Length: 2792 Insert Done parsing in 0.078 seconds

Output:

```

C:\Users\DeLL\OneDrive\Desktop X + v
-----College Linked List-----
Select the operation you want to perform:-
1. Create a Linked List
2. Insertion at Beginning
3. Deletion at End
4. Display
5. Exit
Enter your choice:-7
Invalid Choice!Select the operation you want to perform:-
1. Create a Linked List
2. Insertion at Beginning
3. Deletion at End
4. Display
5. Exit
Enter your choice:-1
Creating a New Linked List
Enter the College ID:-1
Enter the College Name:-VIPS
Enter Number of Students in College:-10000
Select the operation you want to perform:-
1. Create a Linked List
2. Insertion at Beginning
3. Deletion at End
4. Display
5. Exit
Enter your choice:-1
Linked List already exists! Try inserting at beginning.
Select the operation you want to perform:-
1. Create a Linked List
2. Insertion at Beginning
3. Deletion at End
4. Display
5. Exit
Enter your choice:-2
Enter the College ID:-2
Enter the College Name:-BVCOE
Enter Number of Students in College:-8000
Select the operation you want to perform:-
1. Create a Linked List
2. Insertion at Beginning
3. Deletion at End
4. Display
5. Exit
Enter your choice:-4
2
BVCOE

```

```

C:\Users\DeLL\OneDrive\Desktop X + v
Enter Number of Students in College:-8000
Select the operation you want to perform:-
1. Create a Linked List
2. Insertion at Beginning
3. Deletion at End
4. Display
5. Exit
Enter your choice:-4
2
BVCOE
8000
1
VIPS
10000
Select the operation you want to perform:-
1. Create a Linked List
2. Insertion at Beginning
3. Deletion at End
4. Display
5. Exit
Enter your choice:-3
Last Element deleted successfully!
Select the operation you want to perform:-
1. Create a Linked List
2. Insertion at Beginning
3. Deletion at End
4. Display
5. Exit
Enter your choice:-4
2
BVCOE
8000
Select the operation you want to perform:-
1. Create a Linked List
2. Insertion at Beginning
3. Deletion at End
4. Display
5. Exit
Enter your choice:-5
Thanks for using this code!
Invalid Choice!
-----
Process exited after 45.27 seconds with return value 15
Press any key to continue . . .

```

Learning Outcome: