Output:

DOUBLY LINKED LIST

1.) Insert
2.) Insert after a data
3.) Display
4.) Delete
5.) Exit
Enter your choice : 1

Enter info part : 1
Enter your choice : 1.
Enter info part : 2.
Enter your choice : 1.
Enter info part : 3
Enter your choice : 3.
1 → 2 → 3.
Enter your choice : 2.
Enter the number after which you wish to add data : 2.
Enter the info part : 4.
4 → 2 → 1.
Enter your choice : 3.
1 → 2 → 4 → 3
Enter your choice : 4.
Enter the data to be deleted : 4.
Enter yourchoice : 3
1 → 2 → 3.

Student Name ..... Vaibhav Saran ..... Roll No.: .....

Experiment No. ..... 8 ..... Date .....

Objective: To implement a doubly linked list in C language.

Code:

```
#include <stdio.h>
#include <stdlib.h>
Struct node
{ int info;
   Struct node *front, *back;
} *start, *temp, *flag, *p, *q;
void insert ()
{ p = (Struct node *) malloc(sizeof (Struct node));
  if (p == NULL)
  { printf ("\n Failed to allocate memory");
    return;
  }
  printf ("\n Enter the info part: ");
  Scanf ("%d", &p->info);
  p-> front = NULL;
  p-> back = NULL;
  if (start == NULL).
     start = p;
  else.
  { q = start;
     while (q-> front != NULL)
          q = q-> front;
     q-> front = p;
     p-> back = q;
  }.
}
```

```
void ainsert() //to insert a node after a given
{ int ch, x;          // node.
  printf("\nEnter the number after which
          you wish to add data: ");
  scanf("%d", &ch);
  printf("\n Enter the info part: ");
  Scanf("%d", &x);
  p = (struct node *) malloc(sizeof(struct node));
  if(p==NULL)
  { printf("\n Failed to allocate memory");
    return;
  }
  p->info = x;
  p->front = NULL;
  p->back = NULL;
  temp = start;
  flag = start->front;
  while(temp->info != ch)
  { temp = temp->front;
    flag = flag->front;
  }
  if(temp->front==NULL && temp->info!=x)
  { printf("\n Value not found\n");
    return;
  }
  temp->front = p;
  p->back = temp;
  p->front = flag;
  flag->back = p;
  q = p;
```

```
    while (q != NULL)
    { printf("%d->", q->info);
      q = q->back;
    }
    printf("\b\b\b  \n");
}
void delete()
{ int x;
  printf("\n Enter the data to be deleted: ");
  scanf("%d", &x);
  temp = start->front;
  flag = start;
  if (flag->info == x)
  { start = start->front;
    start->back = NULL;
    free(flag);
  }
  while (temp->info != x)
  { temp = temp->front;
    flag = flag->front;
  }
  if (temp->info != x && temp->front == NULL)
  { printf("\n Value does not exist");
    return;
  }
  q = temp->front;
  flag->front = q;
  q->back = flag;
  temp->front = NULL;
  temp->back = NULL;
  free(temp);
```

```
Void display().
{  temp= start;
   while (temp!= NULL).
   {  printf ("%d. -> ", temp -> info);
      temp= temp -> front;
   }
   printf ("\b\b\b   \n");
}
int main ()
{  int ch;
   start = NULL;
   temp= NULL;
   flag =NULL;
   p= NULL;
   q = NULL;
   printf ("\t\t\t\t DOUBLY LINKED LIST \n");
   printf ("1) Insert \n");
   printf ("2) Insert after a data \n");
   printf ("3) Display \n");
   printf ("4) Delete \n");
   Printf ("5) Exit \n");
   printf ("Enter your choice: ");
   Scanf ("%d", &ch);
   do
   {  Switch (ch)
      {  case 1 : insert();
                  break;
         case 2 : printf ("\n\n"); ainsert();
                  break;
```

```
            case 3: printf("\n\n");
                    display();
                    break;
            case 4: delete();
                    break;
            case 5: break;
        }
        printf("\t\t\t\t DOUBLY LINKED LIST \n");
        printf("1) Insert \n");
        printf("2) Insert after a data\n");
        printf("3) Display \n");
        printf("4) Delete \n");
        printf("5) Exit \n");
        printf("Enter your choice: ");
        scanf("%d", &ch);
    }while(ch!=5);
    free(start);
    free(temp);
    free(flag);
    free(p);
    free(q);
    return 0;
}
```