1.Write a program to insert, delete,search and display in Doubly Linked list

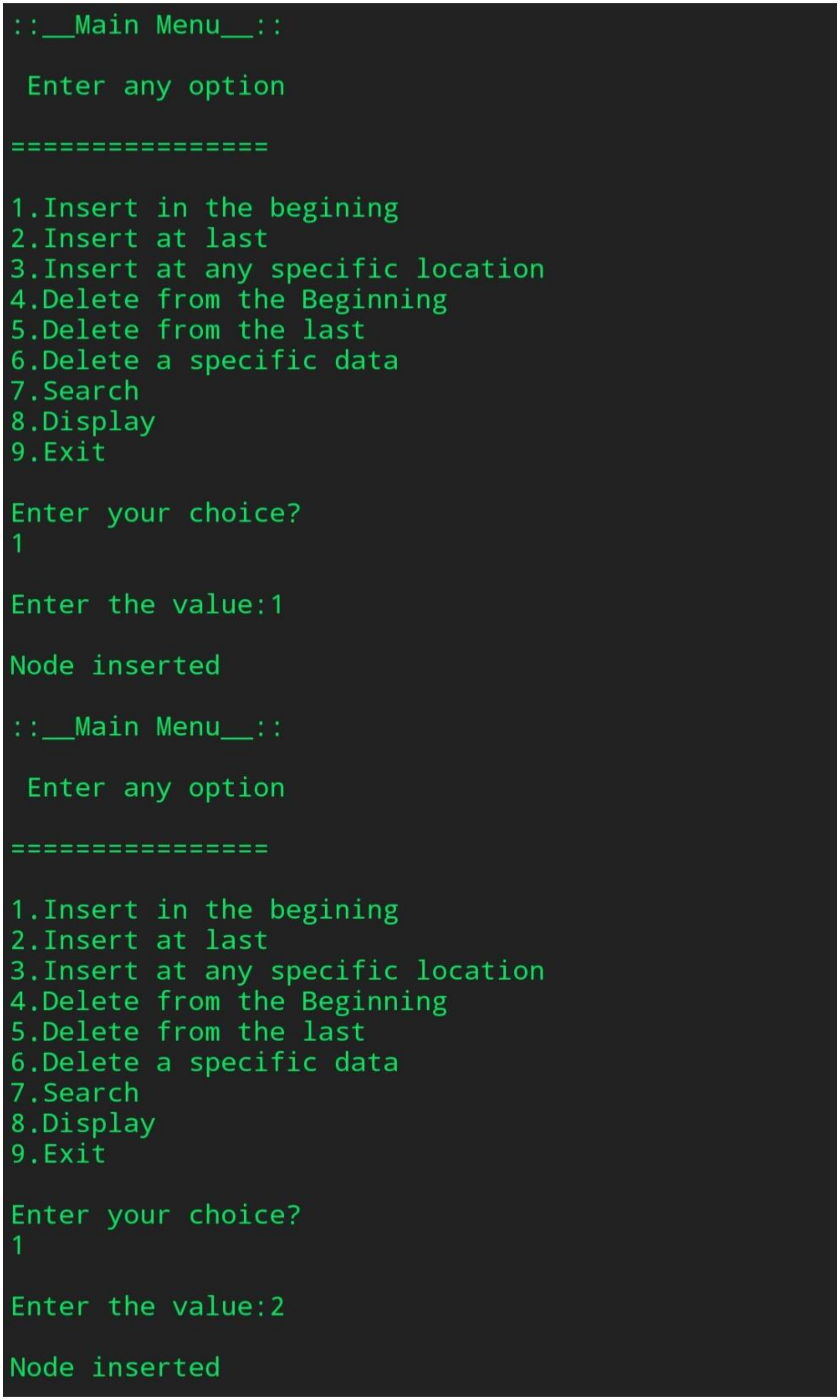
Programs

```
#include<stdio.h>
#include<stdlib.h>
Struct node
{
   Struct node *prev;
   Struct node *next;
   Int data;
};
Struct node *head;
Void insertion_beginning();
Void insertion_last();
Void insertion_specified();
Void deletion_beginning();
Void deletion_last();
Void deletion_specified();
Void display();
Void search();
Void main ()
{
Int choice =1;
   While(choice!= 9)
   {
      Printf("\n::__Main Menu__::\n");
      Printf("\n Enter any option\n");
      Printf("\n================\n");
```

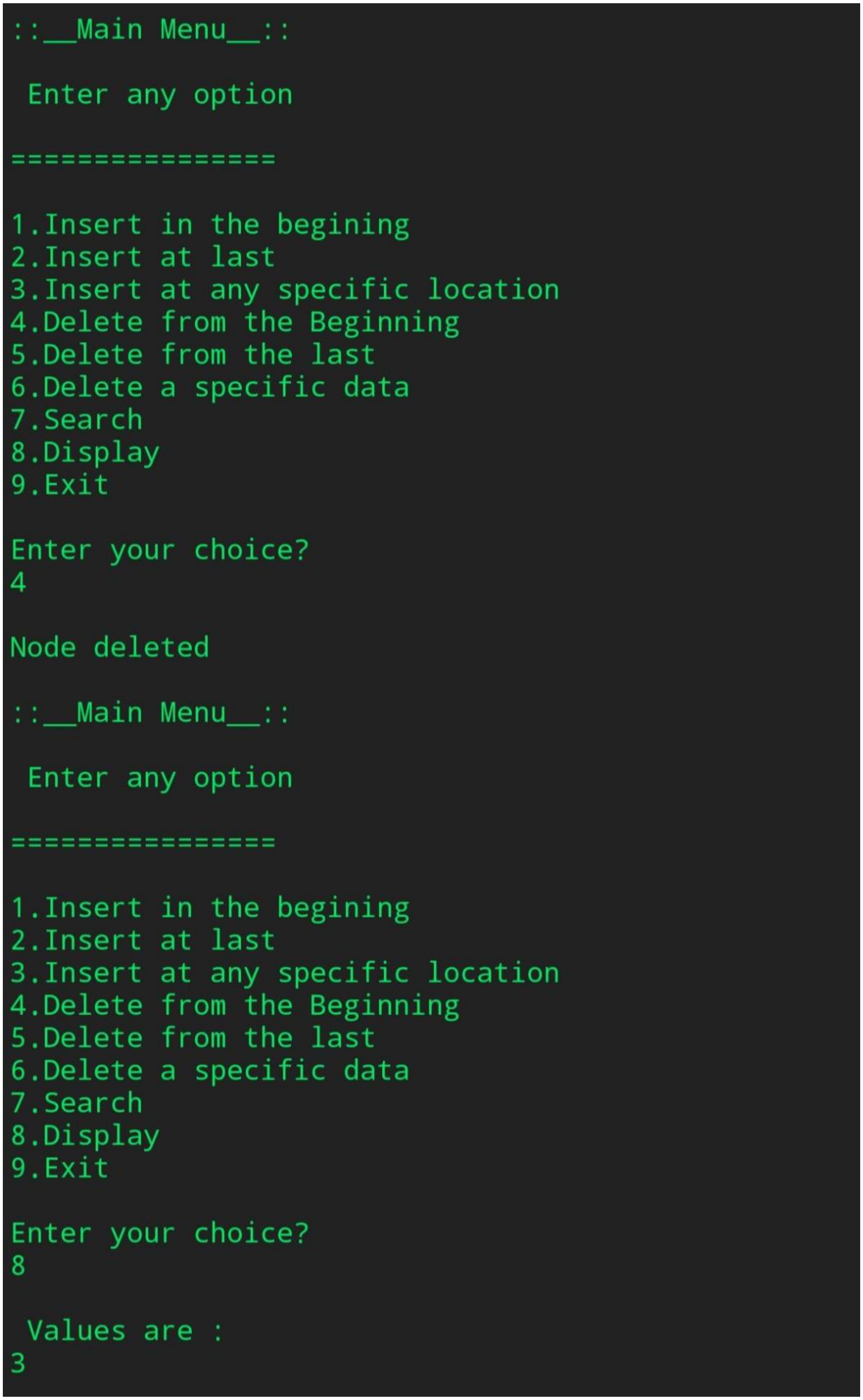
```
    Printf(“\n1.Insert in the beginning\n2.Insert at last\n3.Insert at any specific location\n4.Delete from the Beginning\n5.Delete from the last\n6.Delete a specific data\n7.Search\n8.Display\n9.Exit\n”);

    Printf(“\nEnter your choice?\n”);

    Scanf(“\n%d”,&choice);

    Switch(choice)

    {

      Case 1:

         Insertion_beginning();

      Break;

      Case 2:

          Insertion_last();

      Break;

      Case 3:

      Insertion_specified();

      Break;

      Case 4:

      Deletion_beginning();

      Break;

      Case 5:

      Deletion_last();

      Break;

      Case 6:

      Deletion_specified();

      Break;

      Case 7:

      Search();

      Break;

      Case 8:

      Display();
```

```
          Break;
          Case 9:
          Exit(0);
          Break;
          Default:
          Printf("Invalid choice.Please enter valid choice..");
        }
     }
}
Void insertion_beginning()
{
  Struct node *ptr;
  Int item;
  Ptr = (struct node *)malloc(sizeof(struct node));
  If(ptr == NULL)
  {
     Printf("\nOVERFLOW");
  }
  Else
  {
   Printf("\nEnter the value:");
   Scanf("%d",&item);

  If(head==NULL)
  {
     Ptr->next = NULL;
     Ptr->prev=NULL;
     Ptr->data=item;
     Head=ptr;
  }
  Else
  {
     Ptr->data=item;
     Ptr->prev=NULL;
     Ptr->next = head;
     Head->prev=ptr;
     Head=ptr;
  }
  Printf("\nNode inserted\n");
}

}
Void insertion_last()
{
  Struct node *ptr,*temp;
  Int item;
  Ptr = (struct node *) malloc(sizeof(struct node));
  If(ptr == NULL)
  {
     Printf("\nOVERFLOW");
  }
  Else
  {
     Printf("\nEnter the value:");
     Scanf("%d",&item);
      Ptr->data=item;
     If(head == NULL)
     {
       Ptr->next = NULL;
       Ptr->prev = NULL;
       Head = ptr;
    }
    Else
    {
      Temp = head;
      While(temp->next!=NULL)
      {
         Temp = temp->next;
      }
      Temp->next = ptr;
      Ptr ->prev=temp;
      Ptr->next = NULL;
      }

    }
   Printf("\nNode inserted\n");
  }
Void insertion_specified()
{
  Struct node *ptr,*temp;
  Int item,loc,I;
  Ptr = (struct node *)malloc(sizeof(struct node));
  If(ptr == NULL)
  {
     Printf("\n OVERFLOW");
  }
  Else
  {
    Temp=head;
    Printf(“Enter the location”);
    Scanf(“%d”,&loc);
    For(i=0;i<loc;i++)
    {
      Temp = temp->next;
      If(temp == NULL)
      {
        Printf(“\n There are less than %d elements”, loc);
        Return;
      }
    }
    Printf(“Enter value”);
    Scanf(“%d”,&item);
    Ptr->data = item;
    Ptr->next = temp->next;
    Ptr -> prev = temp;
    Temp->next = ptr;
    Temp->next->prev=ptr;
    Printf(“\nNode inserted\n”);
  }
}
Void deletion_beginning()
{
  Struct node *ptr;
  If(head == NULL)
  {
    Printf(“\n UNDERFLOW”);
    }
    Else if(head->next == NULL)
    {
        Head = NULL;
        Free(head);
        Printf("\nnode deleted\n");
    }
    Else
    {
        Ptr = head;
        Head = head -> next;
        Head -> prev = NULL;
        Free(ptr);
        Printf("\nNode deleted\n");
    }

}
Void deletion_last()
{
    Struct node *ptr;
    If(head == NULL)
    {
        Printf("\n UNDERFLOW");
    }
    Else if(head->next == NULL)
    {
        Head = NULL;
        Free(head);
        Printf("\nNode deleted\n");
    }
    Else
    {
        Ptr = head;
        If(ptr->next != NULL)
        {
            Ptr = ptr -> next;
        }
        Ptr -> prev -> next = NULL;
        Free(ptr);
        Printf("\nNode deleted\n");
    }
}
Void deletion_specified()
{
    Struct node *ptr, *temp;
    Int val;
    Printf("\n Enter the data to be deleted : ");
    Scanf("%d", &val);
    Ptr = head;
    While(ptr -> data != val)
    Ptr = ptr -> next;
    If(ptr -> next == NULL)
    {
        Printf("\nCan't delete\n");
    }
    Else if(ptr -> next -> next == NULL)
    {
        Ptr ->next = NULL;
    }
    Else
    {
        Temp = ptr -> next;
        Ptr -> next = temp -> next;
        Temp -> next -> prev = ptr;
        Free(temp);
        Printf(“\nNode deleted\n”);
    }
}
Void display()
{
    Struct node *ptr;
    Printf(“\n Values are :\n”);
    Ptr = head;
    While(ptr != NULL)
    {
        Printf(“%d\n”,ptr->data);
        Ptr=ptr->next;
    }
}
Void search()
{
    Struct node *ptr;
    Int item,i=0,flag;
    Ptr = head;
    If(ptr == NULL)
    {
        Printf(“\nEmpty List\n”);
    }
    Else
    {
        Printf("\nEnter the data to be searched?\n");
        Scanf("%d",&item);
        While (ptr!=NULL)
        {
            If(ptr->data == item)
            {
                Printf("\nItem found at location %d ",i+1);
                Flag=0;
                Break;
            }
            Else
            {
                Flag=1;
            }
            I++;
            Ptr = ptr -> next;
        }
        If(flag==1)
        {
            Printf("\nItem not found\n");
        }
    }

}
```

OUTPUT



```
::__Main Menu__::

 Enter any option

================

1.Insert in the begining
2.Insert at last
3.Insert at any specific location
4.Delete from the Beginning
5.Delete from the last
6.Delete a specific data
7.Search
8.Display
9.Exit

Enter your choice?
1

Enter the value:1

Node inserted

::__Main Menu__::

 Enter any option

================

1.Insert in the begining
2.Insert at last
3.Insert at any specific location
4.Delete from the Beginning
5.Delete from the last
6.Delete a specific data
7.Search
8.Display
9.Exit

Enter your choice?
1

Enter the value:2

Node inserted
```



```
::__Main Menu__::

 Enter any option

================

1.Insert in the begining
2.Insert at last
3.Insert at any specific location
4.Delete from the Beginning
5.Delete from the last
6.Delete a specific data
7.Search
8.Display
9.Exit

Enter your choice?
4

Node deleted

::__Main Menu__::

 Enter any option

================

1.Insert in the begining
2.Insert at last
3.Insert at any specific location
4.Delete from the Beginning
5.Delete from the last
6.Delete a specific data
7.Search
8.Display
9.Exit

Enter your choice?
8

 Values are :
3
```