// IMPLEMENTATION OF ALL THE OPERATIONS OF DOUBLY LINKED LIST //

ALOGORITHM:

Step 1: if ptr =NULL

Step 2: set new\_node=ptr

Step 3: set ptr=ptr->NEXT

Step 4: set new\_node->data=val

Step 5: set new\_node->prev=NULL

Step 6: set new\_node->next=start

Step 7: set head->prev=new\_node

Step 8 : set head=new node

PROGRAM:

#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()

{

In choice=0;

While(choice !=9)

{

printf(“\n\*\*\*\*\*\*\*\*\*main menu\*\*\*\*\*\*\*\*\*\n”);

printf(“\n choose one option from the following list…\n”);

printf(“\n=================================\n”);

printf(“\n 1. Insert in beginning\n2.insert at last\n3.insert at any random loction\n4.delete from beginning\n5.delete from last\n6.delete the node after the given data\n7.search\n8.show\n9.exit\n”);

printf(“\n enter 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(“please enter vaild choice..”);

}

}

}

Void insertion\_beginning()P

{

Struct node\*ptr;

Int item;

Ptr=(struct node\*)malloc(sizeof(struct node));

If(ptr==NULL)

{

Printf(“\NOVERFLOW”);

}

Else

{

Printf(“\nenter item 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->data=item;

Head=ptr;

}

Printf(“\n node inserted\n”);

}

}

Void insertion\_last()

{

Struct node\*ptr,\*temp;

Int item;

Ptr=(struct node\*)malloc(sizeof(struct node));

If(pt==NULL);

{

Printf(“\Noverflow”);

}

Else

{

Printf(“\n enter 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 element”,loc);

Return;

}

}

Printf(“enter value “);

Scanf(“%d”,&item);

Ptr->data=item;

Ptr->next=temp->next;

Ptr->prev=temp;

Temp->next->prev=ptr;

Printf(“\nnode inserted\n”);

}

Void deletion\_beginning();

{

Struct node\*ptr;

If(head==NULL)

{

Printf(“\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 deletion\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 after which the node is to 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->=NULL;

Ptr->next=temp->next;

Temp->next->prev=ptr;

Free(temp);

Printf(“\nnode deleted\n”);

}

}

Void display()

{

Struct node\*ptr;

Printf(“\n printing values..\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(“\n enter item which you want to search?\n”);

Scanf(“%d”,&item);

While(ptr!=NULL)

{

If(ptr->data==item)

{

Printf(“\n item found at locstion %D”,i+1);\

Flag=0;

Break;

}

Else

{

flag=-1;

}

I++;

Ptr=ptr->next;

}

If(flag==1)

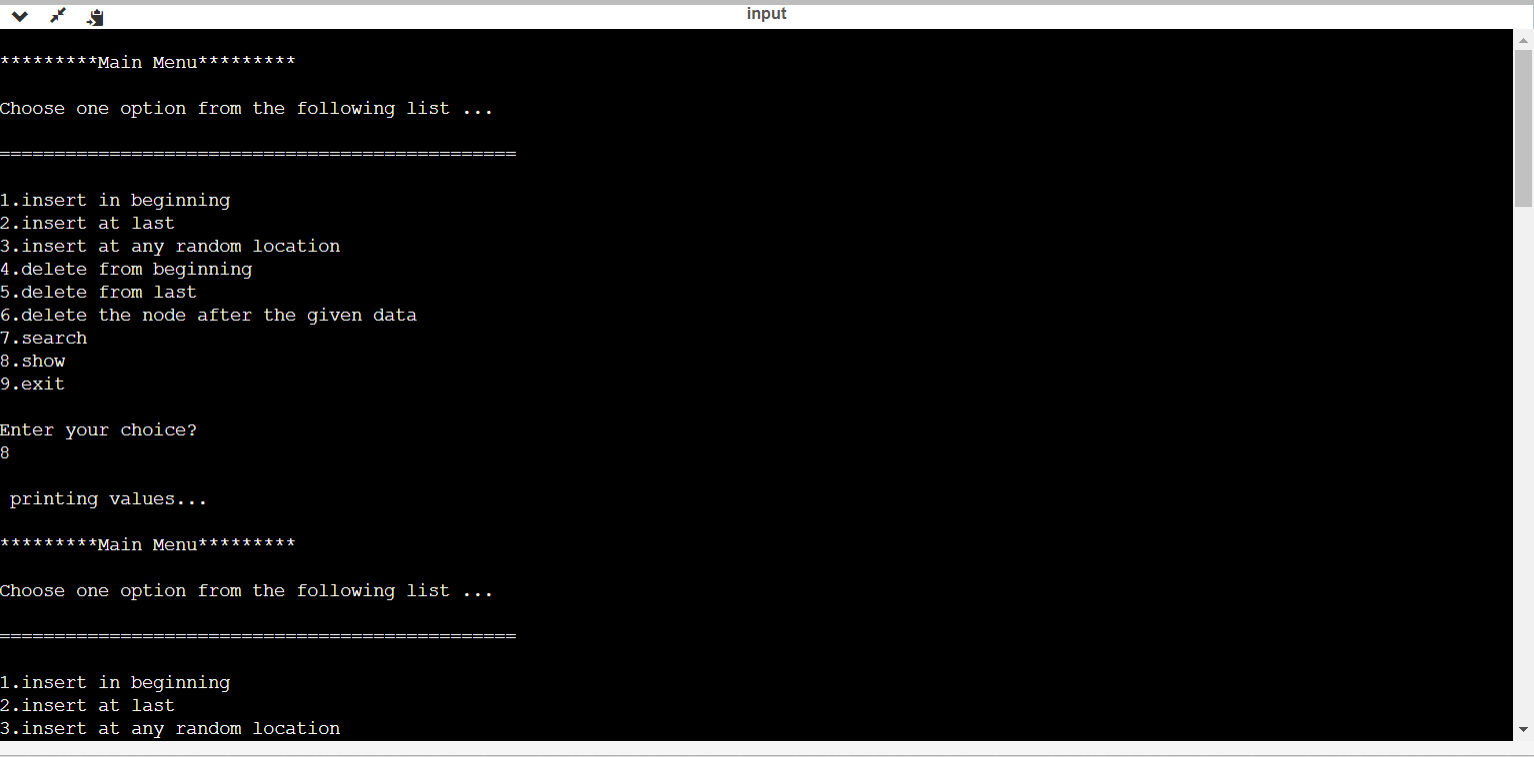
{

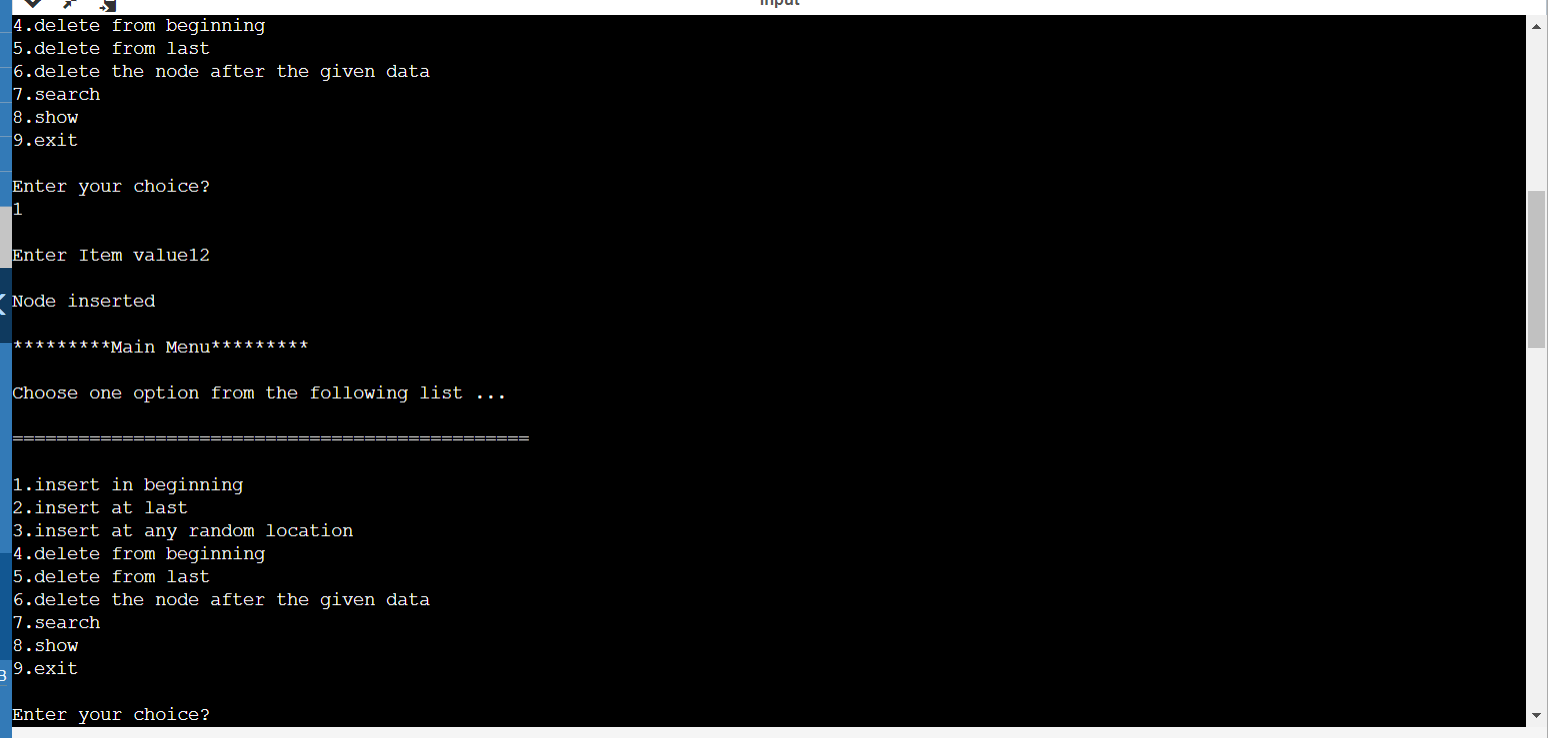
Printf(“\n item not found\n”);

}  
}

}

Out put:





Github link :