DOUBLY LINK LIST

```
#include<stdio.h>
#include<stdlib.h>
struct Node
   {
      int data;
      struct Node* prev;
      struct Node* next;
   };
```

```
void insertEnd(struct Node** head,int key)
  {
      struct Node* current,*parrent;
      struct Node* newnode;
      newnode=(struct Node*)malloc(sizeof(struct Node));
      newnode->data=key;
      newnode->prev=NULL;
      newnode->next=NULL;

      if(*head==NULL)
        {
          *head=newnode;
        }
      else
        {   current=*head;

            while(current)
              {
                parrent=current;
                current=current->next;
              }
            parrent->next=newnode;
            newnode->prev=parrent;
        }
  }
```

```
void Insertbegin(struct Node** head,int key)
  {
     struct Node* current ;
     struct Node *newnode;
     newnode=(struct Node*)malloc(sizeof(struct Node));
     newnode->data=key;
     newnode->prev=NULL;
     newnode->next=NULL;
     printf("you are inside the insertbrgin:\n");
     if((*head)==NULL)
     {
        (*head)=newnode;
     }

     else
     {
        printf("this is insertion part:\n");
       newnode->next=(*head);
       (*head)->prev=newnode;
       (*head)=newnode;
     }
  }

void deletBegin(struct Node** head)
  {
     struct Node* current;
```

```
        if(*head==NULL)
        {
           printf("stack is underflow:\n");
        }
        else if((*head)->next==NULL)
        {
           *head=NULL;
        }
        else if((*head)->next!=NULL)
        {   current=(*head)->next;
           (*head)->next->prev=NULL;
           (*head)->next=NULL;
           *head=current;
        }
        else
           {

           }
     }

void deletEnd(struct Node** head)
        {
           struct Node* current;
           current=*head;
           if(*head==NULL)
           {
              printf("linklist is underflow:\n");
           }
           else if((*head)->next==NULL)
           {
              *head=NULL;

           }
           else
           {

              while(current->next)
              {
                 current=current->next;
              }
              current->prev->next=NULL;
              current->prev=NULL;
           }
        }

void search(struct Node* head)
     {
        struct Node* current;
        int key,count=0,flag=0;
        current=head;
        printf("Enter the value of which you want to
search:\n");
```

```
        scanf("%d",&key);
        if(head==NULL)
        {
            printf("stack is underflow you can not search the
item:\n");
        }
        else if(head!=NULL)
        {
            while(current)
            {
                if(current->data==key)
                {   count++;
                    printf("item is present in the doublelinklist:\n");
                    printf("it is present in node:(%d)\n",count);
                    flag=1;
                }
            }

            if(flag==0)
            {
                printf("item has not been found:\n");
            }
        }
    }

void display(struct Node* head)
```

```
        {
            struct Node* current;
            current=head;
            if(head==NULL)
            {
                printf("double linklist is underflow:\n");
            }

            else
                {   while(current)
                    {
                    printf("%d\t",current->data);
                    current=current->next;
                    }
                }
        }
int main()
    {
        struct Node* head=NULL;
        int choice,key;
        printf("\n*********Stack operations using linked
list*********\n");
        printf("\n----------------------------------------------\n");
    while(choice != 7)
    {
        printf("\n\nChose one from the below options...\n");
printf("\n1.insertBegin\n2.insertEnd\n3.deletBegin\n4.delet
End\n5search.\n6.show\n7.Exiting...");
        printf("\n Enter your choice \n");
        scanf("%d",&choice);
        switch(choice)
        {    case 1:
            {
                printf("enter the value :\n");
                scanf("%d",&key);
                Insertbegin(&head,key);
                break;
            }
            case 2:
            {
                printf("enter the value:\n");
                scanf("%d",&key);
                insertEnd(&head,key);
                break;
            }
            case 3:
            {
                deletBegin(&head);
                break;
            }
            case 4:
```

```
{
   deletEnd(&head);
   break;
}

case 6:
{
   display(head);
   break;
}
 case 5:
{
   search(head);
   break;
}


case 7:
{
   printf("Exiting....");
   break;
}
default:
{
   printf("Please Enter valid choice ");
}
```

```
        };

    }
        return 0;
    }
```