```
//S Harshini-185001058
/*Create a doubly linked list to store set of student names
Perform the operations using a menu driven program */
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include "definition.h"
int main()
{
    node *h,*t;
    char na[20];
    int res,nos;
    h=(node*)malloc(sizeof(node));
    t=(node*)malloc(sizeof(node));
    h->next=t;
    t->prev=h;
    t->next=NULL;
    printf("enter no of students to be entered ");
    scanf("%d",&nos);
    input(h,nos);
    int ch=1;
    while(ch!=0)
    {
        printf("\n Enter choice 1. Insert student name in the front of the list\n 2. Insert
student name at the end of the list \n 3. Insert a record after a given name in the list \n 4. Search
a given student in the list\n 5. Delete a given student \n 6. Display all student names\n 7.
Display the students in alphabetical order \n 8. Enter 0 to exit \n");
        scanf("%d",&ch);
        switch(ch)
        {
        case 1:printf("enter name of student whose name is to be inserted in the front ");
                insertf(h);
                break;
        case 2:printf("enter name of student whose name is to be inserted at last ");
                insertl(t);
                break;
        case 3:printf("enter name of student after whom a given name is to be inserted ");
                scanf("%s",na);
                insert(h,na);
                break;
        case 4:res=search(h);
        if(res==1)
            printf("The student is present");
        else
            printf("The student is not present");
                break;
        case 5:printf("Enter name of student to be deleted ");
                scanf("%s",na);
                delete_stud(h,na);
                break;
        case 6:display(h);
                break;
        case 7:order(h,t);
                break;
        }
    }
    return 0;
}

/* sample input/output

enter no of students to be entered 5
enter the names of students harshu
jaanv
aarthi
goutham
gokhul

 Enter choice 1. Insert student name in the front of the list
 2. Insert student name at the end of the list
 3. Insert a record after a given name in the list
 4. Search a given student in the list
 5. Delete a given student
 6. Display all student names
 7. Display the students in alphabetical order
 8. Enter 0 to exit
1
enter name of student whose name is to be inserted in the front abishek

 Enter choice 1. Insert student name in the front of the list
 2. Insert student name at the end of the list
 3. Insert a record after a given name in the list
 4. Search a given student in the list
 5. Delete a given student
 6. Display all student names
 7. Display the students in alphabetical order
 8. Enter 0 to exit
2
enter name of student whose name is to be inserted at last dilipa

 Enter choice 1. Insert student name in the front of the list
 2. Insert student name at the end of the list
 3. Insert a record after a given name in the list
 4. Search a given student in the list
 5. Delete a given student
 6. Display all student names
 7. Display the students in alphabetical order
 8. Enter 0 to exit
4

 enter name of student to be searched dilipa
The student is present
 Enter choice 1. Insert student name in the front of the list
 2. Insert student name at the end of the list
 3. Insert a record after a given name in the list
 4. Search a given student in the list
 5. Delete a given student
 6. Display all student names
 7. Display the students in alphabetical order
 8. Enter 0 to exit
5
Enter name of student to be deleted dilipa

 Enter choice 1. Insert student name in the front of the list
 2. Insert student name at the end of the list
 3. Insert a record after a given name in the list
 4. Search a given student in the list
 5. Delete a given student
 6. Display all student names
 7. Display the students in alphabetical order
 8. Enter 0 to exit
6

the name of students are abishek
gokhul
goutham
aarthi
jaanv
harshu
```

```
 Enter choice 1. Insert student name in the front of the list
 2. Insert student name at the end of the list
 3. Insert a record after a given name in the list
 4. Search a given student in the list
 5. Delete a given student
 6. Display all student names
 7. Display the students in alphabetical order
 8. Enter 0 to exit
7

 Enter choice 1. Insert student name in the front of the list
 2. Insert student name at the end of the list
 3. Insert a record after a given name in the list
 4. Search a given student in the list
 5. Delete a given student
 6. Display all student names
 7. Display the students in alphabetical order
 8. Enter 0 to exit
6

the name of students are aarthi
abishek
gokhul
goutham
harshu
jaanv

Enter choice 1. Insert student name in the front of the list
 2. Insert student name at the end of the list
 3. Insert a record after a given name in the list
 4. Search a given student in the list
 5. Delete a given student
 6. Display all student names
 7. Display the students in alphabetical order
 8. Enter 0 to exit
0
*/


/************************************CONTENTS OF FILE DEFINITION******************************/
/*typedef struct mynode
{
    char name[20];
    struct mynode *next;
    struct mynode *prev;
}node;
#include "prototype.h"
void insertf(node *h)
{
    node *n;
    n=(node*)malloc(sizeof(node));
    scanf("%s",n->name);
    n->next=h->next;
    n->prev=h;
    h->next->prev=n;
    h->next=n;
}
void input(node *h,int no)
{
    printf("enter the names of students ");
    for(int i=0;i<no;i++)
    {
        insertf(h);
    }
}
void insertl(node *t)
{
    node *p;
    p=(node*)malloc(sizeof(node));
    scanf("%s",p->name);
    p->prev=t->prev;
    p->next=t;
    t->prev->next=p;
    t->prev=p;
}
void insert(node *h,char ch[])
{
    printf("enter name to be inserted ");
    node *tr,*q;
    q=(node*)malloc(sizeof(node));
    scanf("%s",q->name);
    for(tr=h->next;tr!=NULL;tr=tr->next)
    {
        if(strcmp(tr->name,ch)==0)
        {
```

```
			q->prev=tr;
			q->next=tr->next;
			tr->next->prev=q;
			tr->next=q;
		}
	}
}
int search(node *h)
{
	char sear[20];
	printf("\n enter name of student to be searched ");
	scanf("%s",sear);
	node *tr;
	tr=(node*)malloc(sizeof(node));
	for(tr=h->next;tr!=NULL;tr=tr->next)
	{
		if(strcmp(tr->name,sear)==0)
		{
			return 1;
		}
	}
	return 0;
}

void delete_stud(node *h,char del[])
{
	node *tr,*temp;
	for(tr=h->next;tr!=NULL;tr=tr->next)
	{
		if(strcmp(tr->name,del)==0)
		{
			temp=tr;
			tr->prev->next=tr->next;
			tr->next->prev=tr->prev;
			free(temp);
		}
	}
}
void display(node *h)
{
	node *tr;
	tr=(node*)malloc(sizeof(node));
	printf("\nthe name of students are ");
```

```
            for(tr=h->next;tr->next!=NULL;tr=tr->next)
                        printf("%s\n",tr->name);
}
void order(node *h,node *t)
{
  node *p;
  p=(node*)malloc(sizeof(node));
  node *q;
  q=(node*)malloc(sizeof(node));
  char temp[20];
  for(q=h->next;q->next!=NULL;q=q->next)
  {

            for(p=q->next;p!=NULL;p=p->next)
    {
      if(strcmp(q->name,p->name)>0)
                        {

                                    strcpy(temp,q->name);
                                    strcpy(q->name,p->name);
                                    strcpy(p->name,temp);

                        }
    }
  }
}*/
/*****************************CONTENTS OF FILE PROTOTYPE**************************/
/*
void insertf(node *h);
void input(node *h,int no);
void insertl(node *t);
void insert(node *h,char ch[]);
int search(node *h);
void delete_stud(node *h,char del[]);
void display(node *h);
void order(node *h,node *t);
*/
```