**SSN COLLEGE OF ENGINEERING**

**DEPARTMENT OF CSE**

**ASSIGNMENT 4**

**DOUBLY LINKED LIST**

**NAME:S.MADHUMITHA**

**ROLLNO:185001086**

**PROGRAM:**

**FILE1:VARIABLES**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define var\_h

typedef struct my{

char name[30];

struct my\* next;

struct my\* prev;

}node;

#endif

**FILE2:FUNCTIONS**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define func\_h

node\* create(){

node\* h,\*t;

h=(node\*)malloc(sizeof(struct my));

t=(node\*)malloc(sizeof(struct my));

h->next=t;

t->prev=h;

return h;

}

void insertfirst(node\* head,char n[30]){

node\* p=(node\*)malloc(sizeof(struct my));

strcpy(p->name,n);p->prev=head;

p->next=head->next;

head->next->prev=p;

head->next=p;

printf("added");

}

void insertlast(node\* t,char n[30]){

node\* p;

p=(node\*)malloc(sizeof(struct my));

strcpy(p->name,n);p->next=t;

p->prev=t->prev;t->prev->next=p;

t->prev=p;

printf("added");

}

void insertpos(node\* t, char n[30]){

node\* p,\*temp;

char m[30];

int flag=0,i;

temp=t;

p=(node\*)malloc(sizeof(struct my));

while(temp){

if(!(strcmp(temp->name,n))){

flag=1;

break;}

temp=temp->next;

}

if(flag==0)printf("sorry no such name exists");

else if(flag==1){

printf(" name");scanf("%s",m);

strcpy(p->name,m);

p->next=temp->next; p->prev=temp;

temp->next->prev=p;

temp->next=p;

}

}

void search(node\* head,char n[30]){

int flag=0,count=0,i;

node\* temp;temp=head;

while(temp){

count=count+1;

if(!strcmp(temp->name,n)){

flag=1;

break;}

temp=temp->next;

}

if(flag)printf("found at %d",count);

else printf("not found");

}

void delete(node\* head,char n[30]){

node \*g,\*temp; int flag=0;

temp=head;

g=(node\*)malloc(sizeof(struct my));

if(temp==NULL){printf("list empty");return; }

else{

while(temp!=NULL){

if(!(strcmp(n,temp->name))){

flag=1;

break;}

temp=temp->next;

}

g=temp;

temp->prev->next=temp->next;

temp->next->prev=temp->prev;

printf("%s deleted",g->name);

free(g);

if(flag==0)printf("invalid"); return;}

}

void print(node\* hd,node\* t)

{

node\* temp;

for(temp=hd->next;temp->next!=NULL;temp=temp->next){

printf("%s\n",temp->name);

}printf("%s",temp->name);

}

void sort(node\* hd,node\* t){

node\* temp;

char names1[30];

temp=hd->next;

while(temp!=t){

if((temp->name[0])>(temp->next->name[0]) ||(temp->name[0])==(temp->next->name[0]) ){

strcpy(names1,temp->next->name);

strcpy(temp->next->name,temp->name);

strcpy(temp->name,names1);

}

temp=temp->next;}

print(hd,t);

}

#endif

**FILE3:MAIN PROGRAM**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include”var.h”

#include”func.h”

int main(){

node \*l,\*lt;char c='s';

int choice=0,ch=1; char names[30];

l=create();

lt=l->next;

while(ch){

printf("enter name");

scanf("%s",names);

insertlast(lt,names);

printf("\npress 0 to exit");scanf("%d",&ch);

}

c='s';

while(c=='s' ||c=='S'){

printf("MENU\n1.INSERT FIRST\n2.INSERT LAST\n3.INSERT AFTER A NAME\n4.SEARCH\n5.DELETE A RECORD\n6.DISPLAY\n7.DISPLAY SORTED");

scanf("%d",&choice);

switch(choice){

case 1:printf("enter name");

scanf("%s",names);

insertfirst(l,names);break;

case 2:printf("enter name");

scanf("%s",names);

insertlast(lt,names);

break;

case 3:printf("enter name");

scanf("%s",names);

insertpos(l,names);break;

case 4:printf("enter name");

scanf("%s",names);search(l,names);break;

case 5:printf("enter name");

scanf("%s",names);delete(l,names);break;

case 6:print(l,lt);break;

case 7:sort(l,lt);break;

}printf("continue or exit");scanf(" %c",&c);

}

return 0;

}

**OUTPUT:**

enter namehari

added (CREATING DLL)

press 0 to exit1

enter namegeetha

added

press 0 to exit1

enter nameyadhava

added

press 0 to exit1

enter namepriya

added

press 0 to exit0

MENU

1.INSERT FIRST

2.INSERT LAST

3.INSERT AFTER A NAME

4.SEARCH

5.DELETE A RECORD

6.DISPLAY

7.DISPLAY SORTED 1 **(ADDING AT FIRST)**

enter nameankita

addedcontinue or exit s

MENU

1.INSERT FIRST

2.INSERT LAST

3.INSERT AFTER A NAME

4.SEARCH

5.DELETE A RECORD

6.DISPLAY

7.DISPLAY SORTED 2  **(ADDING AT END)**

enter namerekha

addedcontinue or exit s

MENU

1.INSERT FIRST

2.INSERT LAST

3.INSERT AFTER A NAME

4.SEARCH

5.DELETE A RECORD

6.DISPLAY

7.DISPLAY SORTED 3 **(ADDING AFTER A NAME)**

enter namegeetha

namefahadh

continue or exit s

MENU

1.INSERT FIRST

2.INSERT LAST

3.INSERT AFTER A NAME

4.SEARCH

5.DELETE A RECORD

6.DISPLAY

7.DISPLAY SORTED 4 **(SEARCHING FOR A NAME)**

enter namegeetha

found at 4continue or exit s

MENU

1.INSERT FIRST

2.INSERT LAST

3.INSERT AFTER A NAME

4.SEARCH

5.DELETE A RECORD

6.DISPLAY

7.DISPLAY SORTED5 **(DELETING A NAME)**

enter namegeetha

geetha deletedcontinue or exit s

MENU

1.INSERT FIRST

2.INSERT LAST

3.INSERT AFTER A NAME

4.SEARCH

5.DELETE A RECORD

6.DISPLAY

7.DISPLAY SORTED 6 **(DISPLAY DLL)**

ankita

hari

fahadh

yadhava

priya

rekha

continue or exit s

MENU

1.INSERT FIRST

2.INSERT LAST

3.INSERT AFTER A NAME

4.SEARCH

5.DELETE A RECORD

6.DISPLAY

7.DISPLAY SORTED 7  **(DISPLAY SORTED DLL)**

ankita

fahadh

hari

priya

rekha

yadhava continue or exit n