

# ASSignment :- 5

Expt. No. \_\_\_\_\_

Page No. \_\_\_\_\_

Date \_\_\_\_\_

AIM :

1. Write C program that can store student & no. and name in Double linklist and perform following operations.

- Insert
- Display Reverse & Display All
- Delete First & Last
- Insert First & Last
- Delete Specific
- Insert Specific
- Update

Ans

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<process.h>
```

```
struct stud
```

```
{
```

```
    struct stud *ptr;
```

```
    int dno;
```

```
    char name[20];
```

```
    struct stud *next;
```

```
y;
```

```
struct stud *head=NULL, *tmp, *new;
```

```
Void inc()
```

```
{
```

```
    if(head==NULL)
```

```
{
```

```
    head=(struct stud *)malloc(sizeof  
        (struct stud));
```

AIM

```

printf("Enter Roll no. : ");
scanf("-d", &head->dno);
printf("Enter Name : ");
scanf("-s", head->name);

```

$\text{head} \rightarrow \text{nxt} = \text{NULL};$

$\text{head} \rightarrow \text{pV} = \text{NULL};$

y

else

S

$\text{tmp} = \text{head};$

$\text{newz} = (\text{struct stud} *) \text{malloc}(\text{sizeof}(\text{struct stud}));$

```

printf("Enter Roll no. : ");

```

```

scanf("-d", &newz->dno);

```

```

printf("Enter Name : ");

```

```

scanf("-s", newz->name);

```

$\text{newz} \rightarrow \text{nxt} = \text{NULL};$

$\text{while}(\text{tmp} \rightarrow \text{nxt} != \text{NULL})$

$\text{tmp} = \text{tmp} \rightarrow \text{nxt};$

$\text{tmp} \rightarrow \text{nxt} = \text{newz}$

$\text{newz} \rightarrow \text{pV} = \text{tmp};$

y

y

AIM :

```
void disp_all()
```

{

```
    tmp = head;
```

```
    while (tmp != NULL)
```

{

```
        printf("n->s", tmp->name,
```

```
               tmp->name);
```

```
        tmp = tmp->next;
```

}

y

```
Void disp_prev()
```

{

```
    tmp = head;
```

```
    while (tmp->next != NULL)
```

```
        tmp = tmp->next;
```

```
    while (tmp != NULL)
```

{

```
        printf("n->s", tmp->name,
```

```
               tmp->name);
```

```
        tmp = tmp->prev;
```

}

y

```
Void disp()
```

{

```
    int ch;
```

```
    clrscr();
```

AIM

```
if(head == NULL)
```

```
    printf("In Data is not Inserted");
```

```
else
```

```
{
```

```
    printf("1. Display All");
```

```
    printf("2. Display in Reverse Order");
```

```
    printf("Enter your choice : ");
```

```
    scanf("%d", &ch);
```

```
switch(ch)
```

```
{
```

```
case 1 :
```

```
    disp-all();
```

```
    break;
```

```
case 2 :
```

```
    disp-rev();
```

```
    break;
```

```
default :
```

```
    printf("In Invalid choice In");
```

```
    break;
```

```
}
```

```
y
```

```
y
```

AIM

Void del-f()

{

```
tmp = head;
head = head -> next;
free(tmp);
head -> prev = NULL;
```

y

Void del-l()

{

```
tmp = head;
```

```
while (tmp -> next -> next != NULL)
    tmp = tmp -> next;
```

```
free(tmp -> next);
```

```
tmp -> next = NULL;
```

y

void del()

{

```
int ch;
```

```
clrscr();
```

```
if(head == NULL)
```

```
printf("\nData is not Inserted");
```

```
else
```

{

```
printf("\n1. Delete First");
```

```
printf("\n2. Delete Last");
```

AIM

```
printf("In\nEnter your choice:");  
scanf("%d", &ch);
```

```
switch(ch)
```

{

```
case 1:
```

```
    del_f();
```

```
    break;
```

```
case 2:
```

```
    del_l();
```

```
    break;
```

```
default:
```

```
    printf("In\ninvalid choice In");
```

```
    break;
```

y

y

```
void inc_f()
```

{

```
if(head == NULL)  
    inc();
```

```
else
```

{

```
new1 = (struct stud *) malloc(sizeof(  
    struct stud));
```

```
printf("Enter Rollno. : ");  
scanf("%d", &new1->rollno);
```

AIM

```
Printf("Enter name : ");
scanf("-s", &new1->name);
```

$\text{new1} \rightarrow \text{nxt} = \text{head};$

$\text{head} \rightarrow \text{prev} = \text{new1};$

$\text{head} \rightarrow \text{prev} = \text{head} \rightarrow \text{prev};$

$\text{head} \rightarrow \text{prev} = \text{NULL};$

y

y

```
Void inc() {
```

S

```
if (head == NULL)
```

```
inc();
```

```
else
```

S

```
tmp = head;
```

```
new1 = (struct stud *) malloc(sizeof  
        (struct stud));
```

```
Printf("Enter Roll no. : ");
```

```
scanf("-d", &new1->dno);
```

```
Printf("Enter Name : ");
```

```
scanf("-s", &new1->name);
```

$\text{new1} \rightarrow \text{nxt} = \text{NULL};$

```
while (tmp->nxt != NULL)
```

```
tmp = tmp->nxt;
```

Expt. No. \_\_\_\_\_

AIM

 $\text{tmp} \rightarrow \text{next} = \text{newz};$  $\text{newz} \rightarrow \text{prev} = \text{tmp};$ 

3y

y

void inc-ch()

S

int ch;

cls();

printf("Inz. Insert First ");

printf("Inz. Insert Last ");

printf("In In Enter your choice : ");

scanf("%d", &amp;ch);

switch (ch)

S

case 1:

inc-f();

break;

case 2:

inc-l();

break;

default :

printf("In Invalid choice In ");

break;

y

y

Void del - SP()

{

int cnt = 2, node\_number;

class C();

if (head == NULL)

printf("No Data is not Inserted\n");

else

{

printf("Entered node Number : ");

scanf("%d", &node\_number);

if (node\_number == 2)

del\_f();

else

{

tmp = head;

while (cnt != node\_number &&

tmp->next != NULL)

{

tmp = tmp->next;

cnt++;

y

if (cnt == node\_number)

{

tmp->next->prev = tmp->prev;

tmp->prev->next = tmp->next;

free(tmp);

y

AIM

```

else
    printf("Invalid Node
        Number\n");

```

y

y

y

void inc\_af()

s

int cnt = 0, node\_number;

if (head == NULL)

inc();

else

s

printf("Entered Node Number : ");

scanf("%d", &amp;node\_number);

tmp = head;

while (cnt != node\_number &amp;&amp; tmp-&gt;next != NULL)

s

tmp = tmp-&gt;next;

cnt++;

y

if (cnt == node\_number)

s

```

new1 = (struct Stud *) malloc(sizeof
    (struct Stud));

```

```
Ptintfc("Entered Roll no : ");
scanf("-d", &newz->jno);
Ptintfc("Entered Name : ");
scanf("-s", newz->name);
```

```
newz->next = tmp->next;
tmp->next = p2v = newz;
tmp->next = newz;
newz->p2v = tmp;
```

y

else

```
Ptintfc("Invalid node number\n");
```

y

y

```
Void inc-be()
```

S

```
int cnt=1, node-number;
```

```
if(head==NULL)
```

```
inc();
```

else

S

```
Ptintfc("Entered Node Number = ");
scanf("-d", &node-number);
```

```
tmp = head;
```

AIM

while (cnt != node-number &&  
 tmp->next != NULL)

{

tmp = tmp->next;

cnt++;

y

if (cnt == node-number)

{

new1 = (STRUCT STUD \*) malloc(sizeof  
 (STRUCT STUD));

printf("Entered Roll no. : ");

scanf("%d", &new1->rollno);

printf("Entered Name : ");

scanf("%s", new1->name);

new1->next = tmp;

new1->prev = tmp->prev;

tmp->prev->next = new1;

tmp->prev = new1;

y

y

Void inc-spc()

{

int ch;

clrscr();

AIM

`printf("In 1. Insert after give node number");  
 printf("In 2. Insert before give node number");`

`printf("In In Enter your choice : ");  
 scanf("%d", &ch);`

`switch(ch)  
{`

`case 1:`

`inc-af();  
 break;`

`case 2:`

`inc-be();  
 break;`

`default:`

`printf("In Invalid choice\n");  
 break;`

`y`

`y`

`void update()`

`{`

`int foll_number;`

`clsct();`

`if(head == NULL)`

`printf("In Data is not insert\n");`

`else`

`{`

Expt. No. \_\_\_\_\_

AIM

Print("Entered Roll Number to update  
scanf("%d", &roll-number);

tmp = head;

while (tmp->roll != roll-number &&  
tmp->next != NULL)

{

y

tmp = tmp->next;

{

y

if

(tmp->roll == roll-number)

{

Print("Enter Updated Name : ");  
scanf("%s", tmp->name);

y

else

Print("Invalid Roll Number In");

y

Void main()

{

int ch;

clrscr();

while (1)

{

Print("1. Insert ");

Print("2. Display Reverse & Display All ");

Print("3. Delete First & Last ");

AIM

```
printf("In 4. Insert First & Last");  
printf("In 5 Delete specific");  
printf("In 6 Insert specific");  
printf("In 7. Update");  
printf("In 8. Exit In");
```

```
printf("In Enter your choice :");  
scanf("-1-d", &ch);
```

switch(ch){

case 1:

```
inc();  
break;
```

case 2:

```
dispr();  
break;
```

case 3:

```
del();  
break;
```

case 4:

```
inc-ch();  
break;
```

case 5:

```
del-spc();  
break;
```

case 6:

```
inc-spc();  
break;
```

Expt. No. \_\_\_\_\_

AIM

case 7:

Update();

break;

case 0;

exit(0);

default:

printf("In Invalid choice\n");

break;

y

y

getch();

y

2. write c program that can store only numeric data and sort the list according to bubble sort.

Ans

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
```

struct num

s

int fno;

struct num \*nrt;

y;

struct num \*head=NULL, \*tmp, \*mp1, \*ncur1;

AIM

Void inc()

{

if(head == NULL)

{

head = (struct num \*) malloc(sizeof  
(struct num));

printf("In Enter Rollno. : ");

scanf("%d", &amp;head-&gt;dno);

head-&gt;nx = NULL;

}

else

{

tmp = head;

newz = (struct num \*) malloc(sizeof  
(struct num));

printf("In Enter Rollno. : ");

scanf("%d", &amp;newz-&gt;dno);

newz-&gt;nx = NULL;

while (tmp-&gt;nx != NULL)

tmp = tmp-&gt;nx;

tmp-&gt;nx = newz;

y

y

Expt. No. \_\_\_\_\_

AIM

Void disp-all()

{

int num=1;

tmp = head;

while(tmp != NULL)

{

printf("In %d -&gt; %d ", num, tmp-&gt;no),

num++;

tmp = tmp-&gt;nxt;

y

void sort()

{

int i;

for(tmp = head; tmp != NULL; tmp = tmp-&gt;nxt)

{

for(tmp2 = head; tmp2-&gt;nxt != NULL;

tmp = tmp-&gt;nxt)

{

if(tmp1-&gt;no &gt; tmp-&gt;nxt-&gt;no)

{

i = tmp2-&gt;no;

tmp2-&gt;no = tmp1-&gt;nxt-&gt;no;

tmp1-&gt;nxt-&gt;no = i;

y

y

y

AIM:

disp-all();

y

Void main ()

{

int ch;

cls();

while()

{

printf("In2. Insert Record");

printf("In2. Display All ");

printf("In3. Sort Link List ");

printf("In0. Exit ");

printf("Enter your choice : ");

scanf("y-d",&amp;ch);

switch(ch)

{

case 1:

incc();

break;

case 2:

disp-all();

break;

case 3:

sort();

break;

Expt. No. \_\_\_\_\_

AIM

Case 0:

exit(0);

default:

printf("In Invalid choice\n");

break;

y

y

getch();

y

3. write c program that have two different linklist and merge these two lists and display it.

Ans

```
#include<conio.h>
#include<stdio.h>
#include<process.h>
struct num
{
    int dno;
    struct num *next;
};

struct num *head1=NULL, *head2=NULL, *tmp, *new;
Void inc-fc()
{
    if(head1==NULL)
    {
        head1=(struct num *)malloc(sizeof
        (struct num));
    }
}
```

printf("In Enter Rollno : ");

scanf("%d", &head1->dno);

head1->next = NULL;

{

else

{

tmp = head1;

new1 = (struct num \*) malloc(sizeof  
(struct num));

printf("In Enter Rollno : ");

scanf("%d", &new1->dno);

while (tmp->next != NULL)

{

if (tmp->next == head2)  
break;

y

tmp = tmp->next;

y

if (tmp->next == NULL)

{

new1->next = NULL;

tmp->next = new1;

y

else

{

Expt. No. \_\_\_\_\_

AIM

$$+ \text{tmp} \rightarrow \text{next} = \text{new1}$$

$$\text{new1} \rightarrow \text{next} = \text{head};$$

y

y

Void inc(-s())

S

if (head2 == NULL)

S

$$\text{head2} = (\text{struct num} *) \text{malloc}(\text{sizeof}(\text{struct num}));$$

printf("Enter Rollno : ");

scanf("%d", &amp;head2-&gt;fno);

head2-&gt;next = NULL;

y

else

S

tmp = head2;

$$\text{new1} = (\text{struct num} *) \text{malloc}(\text{sizeof}(\text{struct num}))$$

printf("Entered Rollno : ");  
 scanf("%d", &new1->fno);

new1-&gt;next = NULL;

while ( $\text{tmp} \rightarrow \text{next} \neq \text{NULL}$ )

$\text{tmp} = \text{tmp} \rightarrow \text{next};$

y  
 $\text{tmp} \rightarrow \text{next} = \text{newz};$

y  
Void disp-all()

S

int num = z;

$\text{tmp} = \text{headz};$

while ( $\text{tmp} \neq \text{NULL}$ )

S

$\text{printf}(\text{"In-1.d} \rightarrow \text{-1.d"}, \text{num}, \text{tmp} \rightarrow \text{data});$

num ++;

$\text{tmp} = \text{tmp} \rightarrow \text{next};$

y

y  
Void disp-f()

S

int num = 1;

if ( $\text{headz} == \text{NULL}$ )

$\text{printf}(\text{"No data is not Insert into First Link list"});$

else

S

Expt. No. \_\_\_\_\_

AIM

```

tmp = head2;
while (tmp != NULL)
    {

```

```

        if (tmp == head2)
            break;

```

```

        printf("In 2nd -> 1st", num,
               tmp->data);

```

```

        num++;

```

```

        tmp = tmp->next;

```

y

y

3

```
void disp_S()
```

{

```
    int num = 1;
```

```
    if (head2 == NULL)
```

```
        printf("Input data is not inserted into
                second Link list");
```

else

{

```
    tmp = head2;
```

```
    while (tmp != NULL)
```

{

```
        printf("In 1st -> 2nd", num, tmp->data);
```

```
        num++;

```

```
        tmp = tmp->next;

```

y

y

y

AIM

```
void disp()
```

{

```
    int ch;
```

```
    clrscr();
```

```
    if(head1==NULL && head2==NULL)
```

```
        printf("No data is not Inserted");
```

```
    else
```

{

```
        printf("1. Display First Link list");
```

```
        printf("2. Display Second Link list");
```

```
    printf("Enter your choice : ");
```

```
    scanf("%d", &ch);
```

Switch (ch);

{

case 1:

```
    disp_f();
```

```
    break;
```

case 2:

```
    disp_s();
```

```
    break;
```

default :

```
    printf("Invalid choice");
```

```
    break;
```

y

y

y

Expt. No. \_\_\_\_\_

AIM

Void merge()

{

if(head1!=NULL &amp; head2!=NULL)

{

tmp = head1;

while (tmp-&gt;next != NULL)

{

if (tmp-&gt;next == head2)

break;

y

tmp = tmp-&gt;next;

tmp-&gt;next = head2;

display();

y

else

y

printf("In Two Link list is not created");

void main()

{

int ch;

clrscr();

while (1)

{

printf("In 1. Insert in First link list ");

printf("In 2. Insert in second link list ");

printf("In 3. Display ");

```
Ptintf("In 4. Merge List ");
Ptfintf("Id", &ch);
```

Switch (ch) {

S

case 1 :

```
inc - f();
```

```
break;
```

case 2 :

```
inc - s();
```

```
break;
```

case 3 :

```
disp();
```

```
break;
```

case 4 :

```
merge();
```

```
break;
```

case 0 :

```
exit(0);
```

default :

```
Ptintf("In Invalid choice ");
```

```
break;
```

y

y

getch();

y

AIM

4. write c program that can store only numeric data and perform following operations in Double circular linklist

- Insert First & Last
- Delete Last & First

Ans

```
#include<stdio.h>
#include<conio.h>
#include<process.h>
struct stud
{
    struct stud *ptr;
    int dno;
    struct stud *next;
};

struct stud *head=NULL, *tmp, *new, *tmp1;
void inc()
{
    if(head==NULL)
    {
        head=(struct stud *)malloc(sizeof(struct stud));
        printf("Enter Roll no : ");
        scanf("%d", &head->dno);
        head->next=NULL;
        head->ptr=NULL;
    }
}
```

ELSE

{

 $\text{tmp} = \text{head};$  $\text{newz} = (\text{struct stud} *) \text{malloc}(\text{sizeof}(\text{struct stud}));$  $\text{printf("Entered Roll no. : ");}$  $\text{scanf("%d", &newz->dno);}$  $\text{while}(\text{tmp} \rightarrow \text{nxt} != \text{head})$  $\text{tmp} = \text{tmp} \rightarrow \text{nxt};$  $\text{newz} \rightarrow \text{nxt} = \text{head};$  $\text{head} \rightarrow \text{pvr} = \text{newz};$  $\text{head} = \text{head} \rightarrow \text{pvr};$  $\text{head} \rightarrow \text{pvr} = \text{tmp};$  $\text{tmp} \rightarrow \text{nxt} = \text{head};$ 

y

y

Void inc - 1()

S

if ( $\text{head} == \text{NULL}$ )

{

 $\text{head} = (\text{struct stud} *) \text{malloc}(\text{sizeof}(\text{struct stud}));$  $\text{printf("Entered Roll no : ");}$  $\text{scanf("%d", &head->dno);}$

Expt. No. \_\_\_\_\_

AIM

 $\text{head} \rightarrow \text{next} = \text{NULL};$  $\text{head} \rightarrow \text{prev} = \text{NULL};$ 

y

else

{

Ar

 $\text{tmp} = \text{head};$  $\text{newz} = (\text{struct stud} *) \text{malloc}(\text{sizeof}(\text{struct stud}));$  $\text{printf}("Entered Rollno. : ");$  $\text{scanf}("%d", &\text{newz} \rightarrow \text{roll});$  $\text{newz} \rightarrow \text{next} = \text{head};$  $\text{while}(\text{tmp} \rightarrow \text{next} != \text{head})$  $\text{tmp} = \text{tmp} \rightarrow \text{next};$  $\text{tmp} \rightarrow \text{next} = \text{newz};$  $\text{newz} \rightarrow \text{prev} = \text{tmp};$  $\text{newz} \rightarrow \text{next} = \text{head};$  $\text{head} \rightarrow \text{prev} = \text{newz};$ 

y

void inc\_menu()

{

int ch;

 $\text{printf}("Inz. Insert First");$  $\text{printf}("Inz. Insert Last");$

Printfc("In In Entered your choice = ");  
Scnf("r.d", &ch);

switch(ch)  
{

case 1:

inc-f();

break;

case 2:

inc-l();

break;

default:

Printfc("In Invalid choice ");

break;

}

y  
Void disp()

{

tmp = head;

do

{

Printfc("r.d -> ", tmp->data);

tmp = tmp->next;

while (tmp != head);

y

Expt. No. \_\_\_\_\_

AIM:

Void del-f()

{

tmp1 = head;

tmp = head;

while (tmp-&gt;next != head)

tmp = tmp-&gt;next;

Ar

tmp-&gt;next = tmp1-&gt;next;

head = head-&gt;next;

head-&gt;prev = tmp;

free(tmp1);

y

Void del-l()

{

tmp = head;

while (tmp-&gt;next != head)

{ tmp = tmp-&gt;next;

tmp1 = tmp;

y

tmp-&gt;next = head;

head-&gt;prev = tmp;

free(tmp1);

y

AIM

```
void del()
```

{

```
int ch;
```

```
if(head==NULL)
```

```
printf("In Data is not Inserted");
```

```
else
```

{

```
printf("\n1. Delete First");
```

```
printf("\n2. Delete Last");
```

```
printf("\nEnter your choice : ");
```

```
scanf("r%d", &ch);
```

```
switch(ch)
```

{

```
case 1:
```

```
del();
```

```
break;
```

```
case 2:
```

```
del();
```

```
break;
```

```
default:
```

```
printf("Invalid choice");
```

```
break;
```

y

y

y

Expt. No. \_\_\_\_\_

AIM

Void main()

{

clrscr();

while(1)

{

printf("In 1. Insert First &amp; Last");

printf("In 2. Delete First &amp; Last");

printf("In 3. Display ");

printf("In 0. Exit");

printf("In In Enter your choice");

scanf("%d",&amp;ch);

switch(ch)

{

case 1 :

inc-menu();

break;

case 2 :

del();

break;

case 3 :

disp();

break;

default :

printf("In Invalid choice");

break;

Expt. No \_\_\_\_\_

Page No. \_\_\_\_\_

Date \_\_\_\_\_

AIM

y

y

getch();

y