**Stack implementation using single linked list**

#include<stdio.h>

#include<stdlib.h>

#define null 0

struct node

{

int data;

struct node \*link;

};

typedef struct node node;

node \*first;

void main()

{

void insert\_front(),del\_front(),disp();

int ch;

while(1)

{

printf("Main Menu\n");

printf("1:insert\_front\n2:Delete \_Front\n3:display\n4:Exit\n");

printf("Enter your choice\n");

scanf("%d",&ch);

switch(ch)

{

case 1:insert\_front();

break;

case 2:del\_front();

break;

case 3:display();

break;

case 4:exit(0);

}

} }

void insert\_front()

{

node \*p;

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

printf("Enter the data\n");

scanf("%d", &p->data);

p->link=first;

first=p;

}

void display()

{

int cnt=0;

node \*t;

t=first;

while(t)

{

cnt++;

printf("%d\n",t->data);

t=t->link;

}

printf("Total number of nodes=%d\n",cnt);

}

**OUTPUT:**

**Main Menu**

**1:insert\_front**

**2:Delete \_Front**

**3:display**

**4:Exit**

**Enter your choice**

**1**

**Enter the data**

**2**

**Main Menu**

**1:insert\_front**

**2:Delete \_Front**

**3:display**

**4:Exit**

**Enter your choice**

**1**

**Enter the data**

**3**

**Main Menu**

**1:insert\_front**

**2:Delete \_Front**

**3:display**

**4:Exit**

**Enter your choice**

**1**

**Enter the data**

**4**

**Main Menu**

**1:insert\_front**

**2:Delete \_Front**

**3:display**

**4:Exit**

**Enter your choice**

**2**

**Deleted node is 4Main Menu**

**1:insert\_front**

**2:Delete \_Front**

**3:display**

**4:Exit**

**Enter your choice**

**3**

**3**

**2**

**Total number of nodes=2**

**Main Menu**

**1:insert\_front**

**2:Delete \_Front**

**3:display**

**4:Exit**

**Enter your choice**

**Queue implementation using SLL**

#include<stdio.h>

#include<stdlib.h>

#define null 0

struct node

{

int data;

struct node \*link;

};

typedef struct node node;

node \*first;

void main()

{

void insert\_end(), display(), del\_front();

int ch;

while(1)

{

printf("Main Menu\n");

printf("1:insert\_end\n2:Display\n3:delete\_front\n4:Exit\n");

printf("Enter your choice\n");

scanf("%d",&ch);

switch(ch)

{

case 1:insert\_end();

break;

case 2:display();

break;

case 3:del\_front();

break;

case 4:exit(0);

}

}

}

void insert\_end()

{

node \*p,\*r;

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

printf("Enter the data\n");

scanf("%d",&p->data);

if(first==null)

{

first=p;

p->link=null;

}

r=first;

while(r->link!=null)

{

r=r->link;

}

r->link=p;

p->link=null;

}

void display()

{

int cnt=0;

node \*t;

t=first;

while(t)

{

cnt++;

printf("%d\n",t->data);

t=t->link;

}

printf("Total number of nodes=%d\n",cnt);

}

void del\_front()

{

node \*q;

if(first==null)

{

printf("List empty\n");

return;

}

q=first;

printf("Deleted node is %d",q->data);

first=first->link;

free(q);

}

Main Menu

1:insert\_end

2:Display

3:delete\_front

4:Exit

Enter your choice

1

Enter the data

2

Main Menu

1:insert\_end

2:Display

3:delete\_front

4:Exit

Enter your choice

1

Enter the data

3

Main Menu

1:insert\_end

2:Display

3:delete\_front

4:Exit

Enter your choice

2

2

3

Total number of nodes=2

Main Menu

1:insert\_end

2:Display

3:delete\_front

4:Exit

Enter your choice

1

Enter the data

4

Main Menu

1:insert\_end

2:Display

3:delete\_front

4:Exit

Enter your choice

2

2

3

4

Total number of nodes=3

Main Menu

1:insert\_end

2:Display

3:delete\_front

4:Exit

Enter your choice

3

Deleted node is 2Main Menu

1:insert\_end

2:Display

3:delete\_front

4:Exit

Enter your choice

2

3

4

Total number of nodes=2

Main Menu

1:insert\_end

2:Display

3:delete\_front

4:Exit

Enter your choice