#include<stdio.h>

#include<stdlib.h>

struct node

{

int value;

struct node \*link;

};

typedef struct node node1;

void add\_beg();

void add\_end();

void add\_btw();

void del\_beg();

void del\_end();

void del\_btw();

void display();

node1 \*nptr;

node1 \*start=NULL;

node1 \*create();

void main()

{

int ch;

while(1)

{

printf("\nLinked List Operations\n\n1.Add at beginning\n2.Add at end\n3.Add in between\n4.Delete from beginning\n5.Delete from end\n6.Delete in between\n7.Display\n8.Exit\n");

printf("Enter your choice :");

scanf("%d",&ch);

switch(ch)

{

case 1:add\_beg();

break;

case 2:add\_end();

break;

case 3:add\_btw();

break;

case 4:del\_beg();

break;

case 5:del\_end();

break;

case 6:del\_btw();

break;

case 7:display();

break;

case 8:exit(0);

break;

}

}

}

node1 \*create()

{

node1 \*nptr=(node1\*)malloc(sizeof(node1));

if (nptr==NULL)

{

printf("Memory Overflow!");

return 0;

}

else

return nptr;

}

void add\_beg()

{

int val;

node1 \*nptr=create();

printf("Enter the element :");

scanf("%d",&val);

nptr->value=val;

if(start==NULL)

{

start=nptr;

nptr->link=NULL;

}

else

{

nptr->link=start;

start=nptr;

}

}

void add\_end()

{

node1 \*temp,\*nptr=create();

int val;

printf("Enter the element :");

scanf("%d",&val);

nptr->value=val;

nptr->link=NULL;

temp=start;

while(temp->link!=NULL)

{

temp=temp->link;

}

temp->link=nptr;

}

void add\_btw()

{

node1 \*temp,\*nptr=create();

int val,pos,i;

printf("Enter the element :");

scanf("%d",&val);

printf("Enter the position to be inserted :");

scanf("%d",&pos);

nptr->value=val;

nptr->link=NULL;

temp=start;

if(pos==1)

{

nptr->link=start;

start=nptr;

}

else

{

for(i=1;i<pos-1;i++)

{

temp=temp->link;

}

nptr->link=temp->link;

temp->link=nptr;

}

}

void del\_beg()

{

node1 \*temp;

if(start==NULL)

printf("\nList Empty!\n");

else{

temp=start;

start=start->link;

free(temp);

}

}

void del\_end()

{

node1 \*temp,\*prev;

temp=start;

while(temp->link!=NULL)

{

prev=temp;

temp=temp->link;

}

prev->link=NULL;

free(temp);

}

void del\_btw()

{

node1 \*temp,\*prev;

int i,pos;

printf("Enter the position of node to deleted :");

scanf("%d",&pos);

temp=start;

if(pos==1)

start=start->link;

for(i=1;i<pos;i++)

{

prev=temp;

temp=temp->link;

}

prev->link=temp->link;

free(temp);

}

void display()

{

node1 \*temp;

if(start==NULL)

printf("\nList Empty!\n");

temp=start;

while(temp!=NULL)

{

printf("%d",temp->value);

temp=temp->link;

}

}

