#include<stdio.h>

#include<malloc.h>

typedef struct node

{

char p\_name[40],p\_brand[40],p\_type[40],c\_name[40];

int p\_id,p\_size,p\_quantity;

char c\_pno[100];

int p\_price;

struct node \*next;

}node;

node \*head,\*p;

int n;

node\* create(int);

void display(node\*,int n);

node\* insert\_loc(node\*,int loc);

node\* delete\_loc(node \*head,int x);

void main()

{

int ch,loc,x;

do

{

printf("\nSingly linked list operation");

printf("\n1.Create");

printf("\n2.Display");

printf("\n3.Insertion");

printf("\n4.Delete");

printf("\nEnter your choice-");

scanf("%d",&ch);

switch(ch)

{

case 1:

printf("Enter no. of item");

scanf("%d",&n);

head=create(n);

break;

case 2:

display(head,n);

break;

case 3:

head=insert\_loc(head,loc);

n=n+1;

break;

case 4:

head=delete\_loc(head,loc);

n=n-1;

break;

}

}while(ch!=5);

}

node\* create(int n)

{

node \*p;

int i;

head=NULL;

if(head==NULL)

{

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

head->next=NULL;

printf("\tEnter product name:");

scanf("%s",(head->p\_name));

printf("\tEnter product brand:");

scanf("%s",(head->p\_brand));

printf("\tEnter product type:");

scanf("%s",(head->p\_type));

printf("\tEnter product id:");

scanf("%d",&(head->p\_id));

printf("\tEnter product price:");

scanf("%d",&(head->p\_price));

printf("\tEnter product size:");

scanf("%d",&(head->p\_size));

printf("\tEnter customer name:");

scanf("%s",(head->c\_name));

printf("\tEnter customer no.:");

scanf("%s",(head->c\_pno));

p=head;

}

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

{

p->next=(node\*)malloc(sizeof(node));

p=p->next;

printf("\tEnter product name:");

scanf("%s",(p->p\_name));

printf("\tEnter product brand:");

scanf("%s",(p->p\_brand));

printf("\tEnter product type:");

scanf("%s",(p->p\_type));

printf("\tEnter product id:");

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

printf("\tEnter product price:");

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

printf("\tEnter product size:");

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

printf("\tEnter customer name:");

scanf("%s",(p->c\_name));

printf("\tEnter customer no.:");

scanf("%s",(p->c\_pno));

p->next=NULL;

}

return(head);

}

void display(node \*head,int n)

{

int i;

node \*temp;

temp=head;

if(temp==NULL)

{

printf("list is empty");

}

else

{

printf("\n list is::");

printf("\nProduct name");

printf("\tProduct brand");

printf("\tProduct type");

printf("\tProduct id");

printf("\tProduct price");

printf("\tProduct size");

printf("\tCustomer name");

printf("\tCustomer no.");

printf("\n-------------------------------------------------------------------------------------------------------------------");

for(i=0;i<n;i++)

{

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

printf("\t\t%s",temp->p\_brand);

printf("\t\t%s",temp->p\_type);

printf("\t\t%d",temp->p\_id);

printf("\t\t%d",temp->p\_price);

printf("\t\t%d",temp->p\_size);

printf("\t\t%s",temp->c\_name);

printf("\t\t%s",temp->c\_pno);

temp=temp->next;

}

}

}

node\* insert\_loc(node \*head,int loc)

{

node \*p,\*q;

int i;

printf("Enter the new position-");

scanf("%d",&loc);

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

printf("Enter the new data-");

printf("\tEnter product name:");

scanf("%s",(p->p\_name));

printf("\tEnter product brand:");

scanf("%s",(p->p\_brand));

printf("\tEnter product type:");

scanf("%s",(p->p\_type));

printf("\tEnter product id:");

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

printf("\tEnter product price:");

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

printf("\tEnter product size:");

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

printf("\tEnter customer name:");

scanf("%s",(p->c\_name));

printf("\tEnter customer no.:");

scanf("%s",(p->c\_pno));

p->next=NULL;

if(loc==0)

{

p->next=head;

return(p);

}

q=head;

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

{

if(q!=NULL)

{

q=q->next;

}

}

p->next=q->next;

q->next=p;

return(head);

}

node\* delete\_loc(node \*head,int loc)

{

node \*p,\*q;

int i;

printf("\nEnter the location-");

scanf("%d",&loc);

if(loc==1)

{

p=head;

head=head->next;

free(p);

return(head);

}

q=head;

for(i=2;i<loc-1;i++)

{

q=q->next;

}

p=q->next;

q->next=p->next;

free(p);

//n=n-1;

return(head);

}