/\*link list copy,reversal\*/

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

};

struct node\*head;

void beginsert();

void display();

void copy();

void reversal();

int main()

{

int choice=0;

while(choice!=5)

{

printf("\*\*main menu\*\*\n");

printf("choose one option from the following list...\n");

printf("1.insert in begining\n2.display\n3.copy a link list to another list\n4.reverse the link list\n5.exit\n");

printf("enter your choice\n");

scanf("%d",&choice);

switch(choice)

{

case 1:beginsert();

break;

case 2:display();

break;

case 3:copy();

break;

case 4:reversal();

break;

case 5:exit(0);

break;

default:

printf("invalid choice\n");

}

}

}

void beginsert()

{

struct node\*ptr;

int item;

ptr=(struct node\*)malloc(sizeof(struct node\*));

if(ptr==NULL)

{

printf("OVERFLOW\n");

}

else

{

printf("enter value\n");

scanf("%d",&item);

ptr->data=item;

ptr->next=head;

head=ptr;

printf("node inserted\n");

}

}

void display() //traversal

{

struct node\*ptr;

ptr=head;

if(ptr==NULL)

{

printf("nothing to print\n");

}

else

{

printf("printing values...\n");

while(ptr!=NULL)

{

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

ptr=ptr->next;

}

}

}

void copy()

{

struct node\*ptr,\*ptr1;

struct node\*head1;

//head1=new\_node;

ptr=(struct node\*)malloc(sizeof(struct node\*));

head1=(struct node\*)malloc(sizeof(struct node\*));

ptr=head->next;

head1->data=NULL;

ptr1=head1;

while(ptr!=NULL)

{

head1->data=ptr->data;

ptr1->next=head1;

ptr1=head1;

ptr=ptr->next;

}

printf("list is copied\n");

}

void reversal()

{

struct node\*q,\*r,\*s;

q=(struct node\*)malloc(sizeof(struct node\*));

q=head->next;

r=NULL;

s=NULL;

while(q!=NULL)

{

s=r;

r=q;

q=q->next;

r->next=s;

}

head->next=r;

printf("the list is reversed\n");

}

