/\*linked list copy,reversal\*/

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node\*link;

};

struct node\*header;

struct node\*header1;

struct node\*create\_ll(struct node\*);

struct node\*display(struct node\*);

struct node\*copy(struct node\*,struct node\*);

struct node\*reversal(struct node\*);

int main()

{

int choice=0;

while(choice!=5)

{

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

printf("1.create list\n2.display the list\n3.copy the linked list into another linked list\n4.reverse the linked list\n5.exit\n");

printf("enter your choice\n");

scanf("%d",&choice);

switch(choice)

{

case 1:header=create\_ll(header);

break;

case 2:header=display(header);

break;

case 3:header1=copy(header1,header);

break;

case 4:header=reversal(header);

break;

case 5:exit(0);

default:

printf("invalid choice\n");

}

}

}

struct node\*create\_ll(struct node\*header)

{

struct node\*new\_node,\*ptr;

int item;

printf("enter -1 to end\n");

printf("enter the data: \n");

scanf("%d",&item);

while(item!=-1)

{

new\_node=(struct node\*)malloc(sizeof(struct node\*));

new\_node->data=item;

if(header==NULL) //list is empty

{

new\_node->link=NULL;

header=new\_node;

}

else

{

ptr=header;

while(ptr->link!=NULL)

{

ptr=ptr->link;

}

ptr->link=new\_node;

new\_node->link=NULL;

}

printf("enter the data: \n");

scanf("%d",&item);

}

printf("link list is created\n");

return header;

}

struct node\*display(struct node\*header)

{

printf("the linked list is below\n");

struct node\*ptr;

ptr=header;

while(ptr!=NULL) //list is not empty

{

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

ptr=ptr->link;

}

return header;

}

struct node\*copy(struct node\*header,struct node\*header1)

{

struct node\*new\_node;

struct node\*ptr,\*ptr1;

new\_node=(struct node\*)malloc(sizeof(struct node\*));

new\_node->data=NULL;

header1=new\_node;

ptr1=header1;

ptr=header;

while(ptr!=NULL)

{

header1->data=ptr->data;

ptr1->link=header1;

ptr1=header1;

ptr=ptr->link;

}

printf("list is copied\n");

return header1;

}

struct node\*reversal(struct node\*header)

{

struct node\*r,\*s; //here,header=q

r=NULL;

s=NULL;

if(header!=NULL)

{

r=header;

s=header->link;

header=header->link;

r->link=NULL; //make 1st node as last node

}

while(header!=NULL)

{

header=header->link;

s->link=r;

r=s;

s=header;

}

header=r; //to linking out the last node

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

return header;

}

