/\*doubly link list copy\*/

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*rnext;

struct node \*lnext;

};

struct node\*header;

void beginsert();

void display();

void copy();

int main()

{

int choice=0;

while(choice!=4)

{

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.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: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->rnext=header;

header=ptr;

printf("node inserted\n");

}

}

void display() //traversal

{

struct node\*ptr;

ptr=header;

if(ptr==NULL)

{

printf("nothing to print\n");

}

else

{

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

while(ptr!=NULL)

{

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

ptr=ptr->rnext;

}

}

}

void copy()

{

struct node\*ptr,\*ptr1;

struct node\*header1;

//header1=new\_node;

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

header1=(struct node\*)malloc(sizeof(struct node\*));

ptr=header->rnext;

header1->data=NULL;

ptr1=header1;

while(ptr!=NULL)

{

header1->data=ptr->data;

ptr1->rnext=header1;

ptr1=header1;

ptr=ptr->rnext;

}

printf("list is copied\n");

}

