/\*merging of two single linked list\*/

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node\*link;

};

struct node\*header1;

struct node\*header2;

struct node\*headermerge;

struct node\*headercon;

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

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

struct node\*merging(struct node\*);

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

int main()

{

int choice=0;

while(choice!=7)

{

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

printf("1.create 1st list\n2.display 1st list\n3.create 2nd list\n4.display 2nd list\n5.merge 2 lists\n6.display the result of merging\n7.exit\n");

printf("enter your choice\n");

scanf("%d",&choice);

switch(choice)

{

case 1:header1=create\_ll(header1);

break;

case 2:header1=display(header1);

break;

case 3:header2=create\_ll(header2);

break;

case 4:header2=display(header2);

break;

case 5:headermerge=merging(headermerge);

break;

case 6:headermerge=display(headermerge);

break;

case 7: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\*merging(struct node\*headermerge)

{

struct node\*ptr1,\*ptr2;

int temp;

headermerge=concatination(header1,header2,headercon);

ptr1=headermerge;

while(ptr1->link!=NULL)

{

ptr2=ptr1->link;

while(ptr2!=NULL) //there are atleast 2 nodes in the list

{

if(ptr1->data>ptr2->data)

{

temp=ptr1->data;

ptr1->data=ptr2->data;

ptr2->data=temp;

}

ptr2=ptr2->link;

}

ptr1=ptr1->link;

}

printf("list merged\n");

return headermerge;

}

struct node\*concatination(struct node\*header1,struct node\*header2,struct node\*headercon)

{

struct node\*ptr;

ptr=header1;

while(ptr->link!=NULL)

{

ptr=ptr->link;

}

ptr->link=header2;

headercon=header1;

return headercon;

}

  