// Concatenation of 2 link lists

# include <stdio.h>

# include <conio.h>

# include <stdlib.h>

# define NULL 0

struct link

{

int data;

struct link \*next;

};

typedef struct link node;

void create(node \*);

node \*concat(node \*,node \*);

void display(node \*);

int count(node \*);

int main(void)

{

int choice,nodes;

node \*head1,\*head2;

do

{

clrscr();

printf("\n 1. Create first list");

printf("\n 2. Create second list");

printf("\n 3. Concatenate the two lists");

printf("\n 4. Count the no. of nodes present in the final list");

printf("\n 5. Exit");

printf("\n Enter choice...............");

scanf("%d",&choice);

switch(choice)

{

case 1:

clrscr();

printf("\n Enter for first list..............");

printf("\n Enter 0 to stop creating");

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

create(head1);

printf("\n The first list is = ");

display(head1);

printf("\n Press any key...........");

getch();

break;

case 2:

clrscr();

printf("\n Enter for second list..............");

printf("\n Enter 0 to stop creating");

head2=(node \*)malloc(sizeof(node \*));

create(head2);

printf("\n The second list is = ");

display(head2);

printf("\n Press any key...........");

getch();

break;

case 3:

clrscr();

printf("\n Caoncatenation going on.............");

printf("\n\n Concatenation completed.....!!!!!!!!!!");

head1=concat(head1,head2);

printf("\n The full list is -->");

display(head1);

getch();

break;

case 4:

clrscr();

printf("\n The list is --> ");

display(head1);

nodes=count(head1);

printf("\n The no. of nodes present is = %d",nodes);

getch();

break;

case 5:

break;

default:

printf("\n Invalid choice entered.....!!!!!!!!!");

printf("\n\n\n Press any key................");

getch();

break;

}

}while(choice!=5);

return(0);

}

void create(node \*list)

{

printf("\n Input data : ");

scanf("%d",&list->data);

if(list->data==0)

{

list->next=NULL;

}

else{

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

create(list->next);

}

}

void display(node \*list)

{

node \*p;

for(p=list;p->next!=NULL;p=p->next)

printf(" %d ",p->data);

if(list->next==NULL)

{

printf("\n\n Empty list !!!!!!");

}

}

node \*concat(node \*list1,node \*list2)

{

node \*p;

p=list1;

while(p->next->next!=NULL)

p=p->next;

p->next=list2;

return(list1);

}

int count(node \*list)

{

if(list->next==NULL)

return(0);

else

return(1+count(list->next));

}