/\*node count,min element of a link list\*/

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

};

struct node\*header;

void beginsert();

void display();

void node\_count();

void min\_element();

int main()

{

int ch=0;

while(ch!=5)

{

printf("\*\*MAIN MENU\*\*\n");

printf("1.insert nodes\n2.display\n3.count the no. of nodes in a single link list\n4.minimum element from the link list\n5.exit\n");

printf("enter your choice\n");

scanf("%d",&ch);

switch(ch)

{

case 1:beginsert();

break;

case 2:display();

break;

case 3:node\_count();

break;

case 4:min\_element();

break;

case 5:exit(0);

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=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->next;

}

}

}

void node\_count()

{

int count=1;

struct node\*ptr;

ptr=header->next;

while(ptr!=NULL)

{

++count;

ptr=ptr->next;

}

printf("the no. of nodes is:%d\n",count);

}

void min\_element()

{

int min;

struct node\*ptr;

ptr=header->next;

min=ptr->data;

ptr=ptr->next;

while(ptr!=NULL)

{

if(ptr->data<min)

{

min=ptr->data;

}

ptr=ptr->next;

}

printf("the min element of that list is:%d\n",min);

}

