**Name : Disha Y. Supekar**

**Class : SY-02 (H batch)**

**PRN : 202101103154**

***Q. Write a program to sort the elements of a linked list in ascending and descending order.***

#include <stdio.h>

#include <stdlib.h>

struct sort *// creating structure*

{

int number;

struct sort \*next;

};

typedef struct sort node;

node \*create\_list() *// creating linked* list

{

int k,n;

node \*p,\*head;

printf("\n How many elements are there in your list? : ");

scanf("%d",&n);

printf("\n");

for (k=0;k<n;k++)

{

if (k==0)

{

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

p=head;

}

else

{

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

p=p->next;

}

printf(" Enter number at node %d : ",k+1);

scanf("%d",&p->number);

}

p->next=NULL;

return (head);

}

void display (node\*\*head) *//function to display the list*

{

node \*p;

p=head;

printf(" HEAD->");

while(p!=NULL)

{

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

p = p->next;

}

printf("NODE");

}

void ascending (node\* head) *//function to arrange the elements in ascending order*

{

node \*c=head,\*i=NULL;

int t;

while (c!=NULL)

{

i = c->next;

while (i!=NULL)

{

if(c->number > i->number)

{

t = c->number;

c->number = i->number;

i->number = t;

}

i=i->next;

}

c = c->next;

}

}

void descending (node\* head) //function to arrange the elements in descending order

{

node \*c=head,\*i=NULL;

int t;

while (c!=NULL)

{

i = c->next;

while (i!=NULL)

{

if(c->number < i->number)

{

t = c->number;

c->number = i->number;

i->number = t;

}

i=i->next;

}

c = c->next;

}

}

void main()

{

node \*head;

head=create\_list();

printf ("\n Original linked list : \n");

display(head);

ascending(head);

printf("\n\n Linked list in ascending order : \n");

display(head);

descending(head);

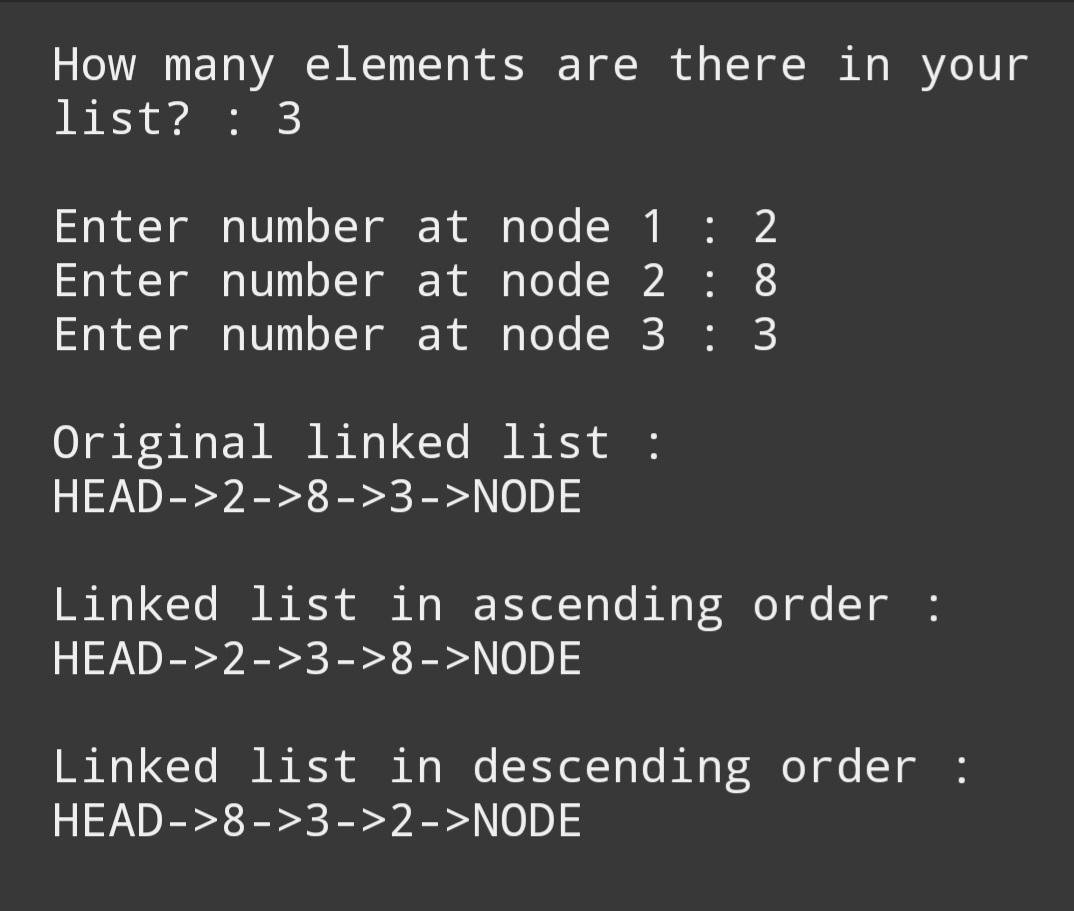
printf("\n\n Linked list in descending order : \n");

display(head);

printf("\n\n");

}

***Output :***



***Q. Write a program to reverse the elements of a singly linked list.***

#include<stdio.h>

#include<stdlib.h>

struct rev *//creating structure*

{

int num;

struct rev \*next;

};

typedef struct rev node;

node\*create\_list()

{

int i,n;

node\*p,\*head;

printf("\n How many elements are there in your list ? : ");

scanf("%d",&n);

printf("\n");

for(i=0;i<n;i++)

{

if(i==0)

{

  head=(node\*)malloc(sizeof(node));

  p=head;

}

else

{

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

  p=p->next;

}

printf(" Enter number at node %d : ",i+1);

scanf("%d",&p->num);

}

p->next=NULL;

return(head);

}

void display(node \*head) *//function to display the list*

{

node \*p;

p=head;

printf("\n HEAD->");

while(p!=NULL)

{

  printf("%d->",p->num);

  p=p->next;

}

printf("NULL");

printf("\n");

}

void reverse(node \*\*head) *//function to reverse the list elements*

{

    node\*c,\*p,\*next;

    c=\*head;

    p=NULL;

    next=NULL;

    while(c!=NULL)

    {

        next=c->next;

        c->next=p;

        p=c;

        c=next;

    }

    \*head=p;

}

void main()

{

node\*head;

head=create\_list();

printf("\n Original linked list : ");

display(head);

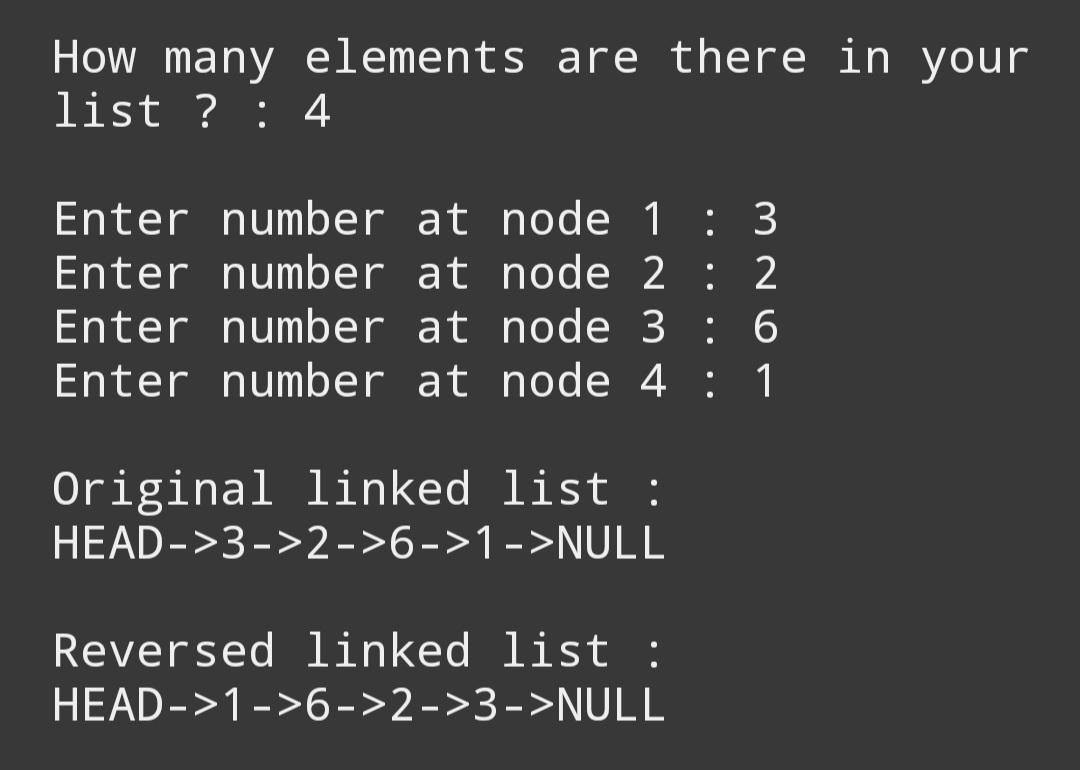
reverse(&head);

printf("\n Reversed linked list : ");

display(head);

}

***Output :***



***Q. Write a program to add elements of two singly linked lists.***

#include <stdio.h>

#include <stdlib.h>

struct Sum

{

int num ;

struct Sum \*next;

};

typedef struct Sum node;

node \*create\_list1()

{

int k,n;

node \*p,\*head1;

printf("\n How many elements are there in list 1 ? : ");

scanf("%d",&n);

for (k=0;k<n;k++)

{

if (k==0)

{

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

p=head1;

}

else

{

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

p=p->next;

}

printf(" Enter number at node %d : ",k+1);

scanf("%d",&p->num);

}

p->next=NULL;

return (head1);

}

node \*create\_list2() // creating linked list

{

int k,n;

node \*p,\*head2;

printf("\n How many elements are there in list 2 ? : ");

scanf("%d",&n);

for (k=0;k<n;k++)

{

if (k==0)

{

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

p=head2;

}

else

{

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

p=p->next;

}

printf(" Enter number at node %d : ",k+1);

scanf("%d",&p->num);

}

p->next=NULL;

return (head2);

}

void display (node\* head)

{

node \*p;

p=head;

printf(" HEAD->");

while(p!=NULL)

{

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

p=p->next;

}

printf("NULL");

}

int add (node\* \*head1,\*head2)

{

node \*a,\*b;

a=head1;

b=head2;

printf("\n\n New list : ");

printf(" HEAD->");

while(a!=NULL && b!=NULL)

{

printf("%d->",(a->num+b->num));

a=a->next;

b=b->next;

}

printf("NULL");

}

void main()

{

node \*head,\*head1,\*head2;

head1=create\_list1();

head2=create\_list2();

printf("\n List 1 : ");

display(head1);

printf("\n\n List 2 : ");

display(head2);

add(head1,head2);

printf("\n\n\n");

}

***Output :***

