#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct node

{

int roll;

char name[20];

node \*next;

};

struct node \*head;

void insert\_node()

{

int r;

char nam[20];

node \*New;

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

printf("Enter roll: ");

scanf("%d",&r);

printf("\nEnter name: ");

scanf("%s",nam);

if(head==NULL)

{

New->roll=r;

strcpy(New->name,nam);

New->next=NULL;

head=New;

}

else

{

int n,m,i;

printf("1: Insert from front\n");

printf("2: Insert from back\n");

printf("3: Insert after specified number of nodes\n");

scanf("%d",&n);

switch(n)

{

case 1:

New->roll=r;

strcpy(New->name,nam);

New->next=head;

head=New;

break;

case 2:

node \*temp;

temp=head;

while(temp->next != NULL)

temp=temp->next;

temp->next=New;

New->roll=r;

strcpy(New->name,nam);

New->next=NULL;

break;

case 3:

printf("\nEnter the node number:");

scanf("%d",&m);

int flag=0;

node \*temp1;

temp1=head;

for(i=1;i<m;i++)

{

temp1=temp1->next;

if(temp1==NULL)

{

printf("\nNode does not exist");

flag=1;

break;

}

}

if(flag==0)

{

New->roll=r;

strcpy(New->name,nam);

New->next=temp1->next;

temp1->next=New;

}

else

break;

}

}

}

void delete\_node()

{

int inp;

int s,i,flag;

if(head==NULL)

{

printf("\nNo node to delete....\n");

}

else

{

node \*t;

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

printf("1: Delete from front: \n");

printf("2: Delete from back: \n");

printf("3: Delete specified number of node: \n");

scanf("%d",&inp);

switch(inp)

{

case 1:

t=head;

head=t->next;

free(t);

break;

case 2:

t=head;

node \*old;

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

if(t->next==NULL)

{

head=NULL;

free(t);

break;

}

else

{

while(t->next!=NULL)

{

old=t;

t=t->next;

}

old->next=NULL;

free(t);

break;

}

case 3:

printf("\nEnter the node number:");

scanf("%d",&s);

t=head;

flag=0;

node \*old1;

old1=(node\*)malloc(sizeof(node));

for(i=1;i<s;i++)

{

old1=t;

if(old1->next==NULL)

{

printf("\nnode does not exist:");

flag=1;

break;

}

t=t->next;

}

if(flag==0)

{

old1->next=t->next;

free(t);

break;

}

else

break;

}

}

}

void display\_node()

{

node \*tt;

if(head==NULL)

printf("\n...NO Item in Link List...\n");

else

{

tt=head;

while(tt!=NULL)

{

printf("\nRoll No: %d",tt->roll);

printf("\nName: %s\n",tt->name);

tt=tt->next;

}

}

}

void sort\_node()

{

node \*tem1;

tem1 = (node\*)malloc(sizeof(node));

node \*tem2;

tem2 = (node\*)malloc(sizeof(node));

int tem = 0;

char nm[20];

for( tem1 = head ; tem1!=NULL ; tem1 = tem1->next )

{

for( tem2 = tem1->next ; tem2!=NULL ; tem2 = tem2->next )

{

if( tem1->roll > tem2->roll )

{

tem = tem1->roll;

strcpy(nm,tem1->name);

tem1->roll = tem2->roll;

strcpy(tem1->name,tem2->name);

tem2->roll = tem;

strcpy(tem2->name,nm);

}

}

}

display\_node();

}

void main()

{

int input;

head=NULL;

while(1)

{

printf("\n-- Menu Selection --\n");

printf("0) Quit\n");

printf("1) Insert\n");

printf("2) Display\n");

printf("3) Sort\n");

printf("4) Delete\n");

//printf("5) Reverse\n");

printf("Enter choice: ");

scanf("%d", &input);

switch(input)

{

case 0:

default:

printf("Goodbye......\n");

exit(0);

break;

case 1:

printf("Insertion.....\n");

insert\_node();

break;

case 2:

printf("Display.....\n");

display\_node();

break;

case 3:

printf("After Sorting.....\n");

sort\_node();

break;

case 4:

printf("Delete....\n");

delete\_node();

break;

}

}

}