//19BEIT30030

//Vaghasiya jayraj

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node \*next;

};

struct node \*head=NULL;

struct stk

{

int stk[20];

int top;

}s;

struct stk1

{

int stk1[20];

int top1;

}s1;

struct queue

{

int que[10];

int front;

int rear;

}q;

void llInsert(int key)

{

struct node \*new;

struct node \*ptr;

new=(struct node\*)malloc(sizeof(struct node));

new->data=key;

new->next=NULL;

ptr=new;

ptr->next=head;

head=ptr;

}

int llDelete()

{

struct node \*ptr,\*temp;

int ans;

ptr=head;

while(ptr->next!=NULL)

{

ptr=ptr->next;

}

ans=ptr->data;

ptr=head;

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

{

temp=ptr->next;

ptr=ptr->next;

}

ptr->next=NULL;

temp=temp->next;

free(temp);

return ans;

}

int llLength()

{

int count=0;

struct node \*ptr,\*temp;

ptr=head;

while(ptr!=NULL)

{

ptr=ptr->next;

count++;

}

return count;

}

void llDisplay()

{

struct node \*ptr;

ptr=head;

while(ptr!=NULL)

{

printf("\t%d",ptr->data);

ptr=ptr->next;

}

}

void init(struct stk \*a,struct queue \*b,struct stk1 \*c)

{

a->top=-1;

c->top1=-1;

b->front=-1;

b->rear=-1;

}

void skPush(struct stk \*a, int data)

{

if(a->top==19)

{

printf("\n=========Stack Overflow=========\n");

}

else

{

a->top++;

a->stk[a->top]=data;

}

}

void skDisplay(struct stk \*a)

{

for(int i=a->top;i>=0;i--)

{

printf("\n\t%d",a->stk[i]);

}

}

void sk1Push(struct stk1 \*c, int data)

{

if(c->top1==19)

{

printf("\n=========Stack Overflow=========\n");

}

else

{

c->top1++;

c->stk1[c->top1]=data;

}

}

void sk1Display(struct stk1 \*c)

{

for(int i=c->top1;i>=0;i--)

{

printf("\n\t%d",c->stk1[i]);

}

}

void enqueue(struct queue \*b,int data)

{

if(b->front==-1)

{

b->front=0;

}

b->rear=(b->rear+1)%10;

b->que[b->rear]=data;

}

int dequeue(struct queue \*b)

{

int data=b->que[b->front];

if(b->front==b->rear)

{

b->front = -1;

b->rear = -1;

}

else

{

b->front=(b->front+1)%10;

}

return data;

}

void quDisplay(struct queue \*b)

{

if(b->rear>=b->front)

{

for(int i=b->front;i<=b->rear;i++)

{

printf("\t%d",b->que[i]);

}

}

else

{

for(int i=b->front;i<10;i++)

{

printf("\t%d",b->que[i]);

}

for(int i=0;i<=b->rear;i++)

{

printf("\t%d",b->que[i]);

}

}

}

void enterDataStkEven()

{

int userData;

printf("\n=========LINKED-LIST-DATA=========\n");

llDisplay();

printf("\n");

printf("\nEnter The Data For PipeLine, press 00 to quit:- ");

scanf("%d",&userData);

if(userData==00)

{

exit(1);

}

llInsert(userData);

int llData=llDelete();

int queData=dequeue(&q);

enqueue(&q,llData);

skPush(&s,queData);

printf("\n=========STACK-2-DATA=========\n\n");

skDisplay(&s);

printf("\n");

printf("\n=========QUEUE-DATA=========\n\n");

quDisplay(&q);

printf("\n");

}

void enterDataStkOdd()

{

int userData;

printf("\n=========LINKED-LIST-DATA=========\n");

llDisplay();

printf("\n");

printf("\nEnter The Data For PipeLine, press 00 to quit:- ");

scanf("%d",&userData);

if(userData==00)

{

exit(1);

}

llInsert(userData);

int llData=llDelete();

int queData=dequeue(&q);

enqueue(&q,llData);

sk1Push(&s1,queData);

printf("\n=========STACK-1-DATA=========\n");

sk1Display(&s1);

printf("\n");

printf("\n=========QUEUE-DATA=========\n");

quDisplay(&q);

printf("\n");

}

void main()

{

int llRange,count=1;

char choice="p";

llInsert(0);

llInsert(1);

llInsert(2);

llInsert(3);

llInsert(4);

printf("\n=========LINKED-LIST-DATA=========\n");

llDisplay();

init(&s,&q,&s1);

enqueue(&q,20);

enqueue(&q,52);

enqueue(&q,85);

enqueue(&q,63);

enqueue(&q,78);

enqueue(&q,56);

enqueue(&q,96);

enqueue(&q,30);

enqueue(&q,29);

enqueue(&q,80);

printf("\n=========QUEUE-DATA=========\n");

quDisplay(&q);

printf("\n");

printf("\n");

printf("\nPipeLine Begins.........");

printf("\n");

while(choice!="Q")

{

if(count%2==0)

{

enterDataStkEven();

count++;

}

else

{

enterDataStkOdd();

count++;

}

}

}