**R.Harini**

**18BCE1010**

#include<stdio.h>

#include<stdlib.h>

#define MAX 10

struct Books{

char accno[10];

char title[20];

char author[30];

};

struct Stack{

int top;

struct Books\* a[50];

};

int is\_empty(struct Stack\* s){

if (s->top==-1){

return 1;

}

else{

return 0;

}

}

int is\_full(struct Stack\* s){

if (s->top==(MAX-1)){

return 1;

}

else{

return 0;

}

}

void push(struct Stack\* s, struct Books\* b){

if (is\_full(s)){

printf("Stack is full\n");

}

else{

s->top++;

s->a[s->top]=b;

}

}

struct Books\* pop(struct Stack\* s){

if (is\_empty(s)){

printf("Stack is empty\n");

}

else{

struct Books\* b=(struct Books\*)malloc(sizeof(struct Books));

b=s->a[s->top];

s->top--;

return b;

}

}

void display\_s(struct Stack\* s){

printf("Displaying the stack:\n");

for (int i=0;i<s->top+1;i++){

struct Books\* b=(struct Books\*)malloc(sizeof(struct Books));

b=s->a[i];

printf("%s\n%s\n%s\n",b->accno,b->title,b->author);

}

}

struct Queue{

int front;

int rear;

struct Books\* a[30];

};

int qis\_full(struct Queue\* q){

if (q->front==q->rear && q->rear>-1){

return 1;

}

else{

return 0;

}

}

int qis\_empty(struct Queue\* q){

if (q->rear==-1){

return 1;

}

else{

return 0;

}

}

void enqueue(struct Queue\* q, struct Books\* b){

if (qis\_full(q)){

printf("Queue is full\n");

}

else{

q->rear++;

q->a[q->rear]=b;

}

}

void dequeue(struct Queue\* q){

if (qis\_empty(q)){

printf("Queue is empty\n");

}

else{

q->front++;

struct Books\* b=(struct Books\*)malloc(sizeof(struct Books));

b=q->a[q->front];

printf("%s\n%s\n%s", b->accno, b->title, b->author);

}

}

void display\_q(struct Queue\* q){

if (qis\_empty(q)){

printf("Queue is empty\n");

}

else{

printf("Displaying the queue:\n");

for (int i=(q->front+1);i<(q->rear+1);i++){

struct Books\* b=(struct Books\*)malloc(sizeof(struct Books));

b=q->a[i];

printf("%s\n%s\n%s\n",b->accno, b->title, b->author);

}

}

}

void separate(struct Stack\* s,struct Queue\* q1, struct Queue\* q2){

for (int i=0;i<s->top+1;i++){

struct Books\* b=(struct Books\*)malloc(sizeof(struct Books));

b=pop(s);

if (b->accno[0]=='C'){

enqueue(q1,b);

//display\_q(q1);

}

else{

enqueue(q2,b);

//display\_q(q2);

}

}

}

int main(){

struct Stack\* c1=(struct Stack\*)malloc(sizeof(struct Stack));

struct Stack\* c2=(struct Stack\*)malloc(sizeof(struct Stack));

c1->top=-1;

c2->top=-1;

struct Queue\* cs=(struct Queue\*)malloc(sizeof(struct Queue));

struct Queue\* ec=(struct Queue\*)malloc(sizeof(struct Queue));

cs->front=-1;

cs->rear=-1;

ec->front=-1;

ec->rear=-1;

int c;

int tot=0;

printf("Enter which counter(1 or 2):");

scanf("%d",&c);

struct Books b1[20];

struct Books b2[20];

while (c!=0){

int n;

if (c==1){

printf("Enter how many books to be returned:");

scanf("%d",&n);

tot+=n;

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

printf("Enter accno, title and author:");

scanf("%s %s %s", b1[i].accno, b1[i].title, b1[i].author);

push(c1,&b1[i]);

display\_s(c1);

}

printf("Enter which counter(1 or 2):");

scanf("%d",&c);

}

else{

printf("Enter how many books to be returned:");

scanf("%d",&n);

tot+=n;

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

printf("Enter accno, title and author:");

scanf("%s %s %s", b2[i].accno, b2[i].title, b2[i].author);

push(c2,&b2[i]);

display\_s(c2);

}

printf("Enter which counter(1 or 2):");

scanf("%d",&c);

}

}

printf("\nTotal number of books returned: %d\n",tot);

separate(c1, cs, ec);

separate(c2, cs, ec);

display\_q(cs);

display\_q(ec);

}