/\*dequeue ,enqueue,display using circular queue\*/

#include<stdio.h>

#include<stdlib.h>

#define MAX 10

int front=0; //we will start the index from 1

int rear=0;

int queue[MAX];

void insert\_queue();

void delete\_queue();

void display\_queue();

int main()

{

int ch;

while(1)

{

printf("\*\*main menu\*\*\n");

printf("press 1 for insert element\n");

printf("press 2 for delete element\n");

printf("press 3 for display\n");

printf("press 4 for exit\n");

printf("enter your choice\n");

scanf("%d",&ch);

switch(ch)

{

case 1:insert\_queue();

break;

case 2:delete\_queue();

break;

case 3:display\_queue();

break;

case 4:exit(0);

default:

printf("invalid choice\n");

}

}

}

void insert\_queue()

{

int item;

if(((rear%MAX)+1)==front)

{

printf("queue is full\n");

}

else

{

if(front==0 && rear==0)

{

front=1;

}

rear=(rear%MAX)+1;

printf("enter a value to be inserted\n");

scanf("%d",&item);

queue[rear]=item;

}

printf("the value is inserted\n");

}

void delete\_queue()

{

int x; //here,item=x

if(front==0)

{

printf("queue is empty\n");

}

else

{

x=queue[front];

if(front==rear) //queue has only one element

{

front=0;

rear=0;

printf("only element of this queue is deleted\n");

}

else

{

front=(front%MAX)+1;

printf("%d is deleted\n",x);

}

}

}

void display\_queue()

{

printf("the queue is below\n");

int i;

if(rear==0 && front==0)

{

printf("queue is empty\n");

}

else

{

for(i=front;i<=rear;i++)

{

printf("%d\n",queue[i]);

}

}

}

