#include <stdio.h>

#include<stdlib.h>

#define MAX\_SIZE 6

int queue[MAX\_SIZE];

int front = -1, rear = -1;

int isFull()

{

return (rear + 1) % MAX\_SIZE == front;

}

int isEmpty()

{

return front == -1;

}

void enqueue(int data)

{

if (isFull())

{

printf("Queue overflow\n");

return;

}

if (front == -1)

{

front = 0;

}

rear = (rear + 1) % MAX\_SIZE;

queue[rear] = data;

printf("Element %d inserted\n", data);

}

int dequeue()

{

if (isEmpty())

{

printf("Queue underflow\n");

return -1;

}

int data = queue[front];

if (front == rear)

{

front = rear = -1;

}

else

{

front = (front + 1) % MAX\_SIZE;

}

return data;

}

void display()

{

if (isEmpty())

{

printf("Queue is empty\n");

return;

}

printf("Queue elements: ");

int i = front;

while (i != rear)

{

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

i = (i + 1) % MAX\_SIZE;

}

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

}

void search()

{

int s,f=0;

int i=front;

printf("Enter element to be search:");

scanf("%d",&s);

while(i!=rear+1)

{

if(s==queue[i])

f=1;

i = (i + 1) % MAX\_SIZE;

}

if(f==0)

{

printf("Element not found\n");

}

else

{

printf("Element found\n");

}

}

int main()

{

int data,ch;

while(1)

{

printf("\nOPERATIONS ON CIRCULAR QUEUE \n 1.Insertion \n 2.Deletion \n 3.Display \n 4.Search \n 5.Exit");

printf("\nEnter your choice:");

scanf("%d",&ch);

switch(ch)

{

case 1:

printf("\nEnter element to be inserted:");

scanf("%d",&data);

enqueue(data);

display();

break;

case 2:

printf("Dequeued element: %d\n", dequeue());

display();

break;

case 3:

display();

break;

case 4:

search();

break;

case 5:

printf("EXIT\n");

exit(0);

break;

default:

printf("\nInvalid entry...try again\n");

break;

}

}

return 0;

}