#include <iostream>

using namespace std;

int dequeue(int queue[], int &front, int &back, int size);

int enqueue(int queue[], int &front, int &back, int size);

int display(int queue[], int &front, int &back, int size);

int element;

int main()

{

int size;

int front =-1;

int back =-1;

cout<<"Enter size of circular queue to initialize: ";

cin>>size;

int queue[size];

int cont=0;

int op=0;

int element;

cout<<"Enter 1 to add to queue\nEnter 2 to delete from queue\nEnter 3 to display queue\n";

while (cont==0)

{

cout<<"Select operation to perform: ";

cin>>op;

switch(op)

{

case 1:

enqueue(queue, front, back, size);

break;

case 2:

dequeue(queue, front, back, size);

break;

case 3:

display(queue, front, back, size);

break;

}

cout<<"Enter 0 to continue, 1 to exit: ";

cin>>cont;

}

return 0;

}

int enqueue(int queue[], int &front, int &back, int size)

{

if((front==0 && back==size-1) || (front==back+1))

{

cout<<"Error: Queue is full\n";

}

else

{

if(front==-1)

front=0;

cout<<"enter element to enqueue: ";

cin>>element;

back=(back+1)%size;

queue[back]=element;

}

return 0;

}

int dequeue(int queue[], int &front, int &back, int size)

{

if(front==-1)

{

cout<<"Error: Queue is empty\n";

}

else

{

if(front==back)

{

front=-1;

back=-1;

cout<<"Element was deleted successfully\n";

}

else

{

front=(front+1)%size;

cout<<"Element was deleted successfully\n";

}

}

return 0;

}

int display(int queue[], int &front, int &back, int size)

{

if(front==-1)

{

cout<<"Error: Queue is empty\n";

}

else

{

int i=front;

for(int j=0; j<size; j++)

{

cout<<queue[i]<<" ";

if(i==back)

break;

i=(i+1)%size;

}

cout<<endl;

}

return 0;

}