#include<iostream>

using namespace std;

class interLeave {

int \*arr;

int size;

int front;

int rear;

public:

interLeave() {

size = 100001;

arr = new int[size];

front = -1;

rear = -1;

}

~interLeave() {

delete[] arr;

}

bool isEmpty() {

if(front == -1 || front > rear) {

return true;

}

else {

return false;

}

}

bool isFull() {

if(rear == size-1) {

return true;

}

else {

return false;

}

}

void enqueue(int data) {

if(isFull()) {

cout<<"OVERFLOW"<<endl;

}

else if(isEmpty()) {

front = rear = 0;

arr[rear] = data;

}

else {

rear++;

arr[rear] = data;

}

}

int dequeue() {

if(isEmpty()) {

cout<<"UNDERFLOW"<<endl;

return -1;

}

else {

// cout << "Dequeued element: " << arr[front] << endl;

int dequeuedvalue = arr[front];

front++;

return (dequeuedvalue);

}

}

void display() {

if(isEmpty()) {

cout<<"Queue is EMPTY!!"<<endl;

}

cout<<"Queue elements: ";

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

cout<<arr[i]<<" ";

}

cout<<endl;

cout<<"Size of Queue : "<<(rear - front + 1)<<endl;

}

};

int main() {

interLeave q1,q2;

int n;

cout<<"Enter size of queue: ";

cin>>n;

cout<<"Enter elements of queue: ";

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

int x;

cin>>x;

q2.enqueue(x);

}

for(int i=0; i<n/2; i++) {

q1.enqueue(q2.dequeue());

}

cout<<"Interleaved Queue: ";

while(!q1.isEmpty() && !q2.isEmpty()) {

cout<<q1.dequeue()<<" "<<q2.dequeue()<<" ";

}

cout<<endl;

return 0;

}

