```cpp
// Name: Hoang Thi Diep
// Purpose: Cai dat va test lop hang doi cac so nguyen
// dung mang vong cap phat dong
// Note: chuong trinh chua dinh nghia ham enqueue va dequeue
// ------------------------------------------------------------------------
#include <iostream>
using namespace std;

#define INIT_CAPACITY 2

// ------------------------------------------------------------------------
// lop ngoai le hang doi rong
class EmptyQueueException{
public:
    string getMessage(){ return "Loi: Hang doi rong!";};
};

// ------------------------------------------------------------------------
// lop hang doi cai dat boi mang vong cap phat dong
// element[r] khong luu du lieu
// => mang chua duoc toi da (capacity - 1) phan tu
class ArrayQueue{
public:
    ArrayQueue();// ham kien tao
    ~ArrayQueue();// ham huy
    int size();// ham dem so phan tu trong hang doi
    bool isEmpty();// ham kiem tra hang doi rong
    int front() throw (EmptyQueueException);// ham tra ve gia tri phan tu o dau hang doi
    int rear() throw (EmptyQueueException);// ham tra ve gia tri phan tu o cuoi hang doi
    void enqueue(int x);// ham them x vao cuoi hang doi
    int dequeue() throw (EmptyQueueException);// ham loai phan tu o dau hang doi
    void print();// ham in hang doi
private:
    bool isFull();// ham kiem tra xem mang day hay chua
    int * element;// con tro toi dau mang
    int capacity;// kich thuoc toi da cua mang
    int f;// chi so cua o dau hang doi
    int r;// chi so cua o lien sau o cuoi hang doi
};

// ham kien tao
ArrayQueue::ArrayQueue(){
    capacity = INIT_CAPACITY;
    element = new int[capacity];
    f = 0;
    r = 0;
}

// ham huy
ArrayQueue::~ArrayQueue(){
    delete [] element;
    element = NULL;
    capacity = 0;
    f = 0;
    r = 0;
}

// ham dem so phan tu trong hang doi
int ArrayQueue::size(){
    return (capacity + r - f) % capacity;
```

```
 61: }
 62:
 63: // ham kiem tra hang doi rong
 64: bool ArrayQueue::isEmpty(){
 65:     return (f == r);
 66: }
 67:
 68: // ham kiem tra xem mang day hay chua
 69: bool ArrayQueue::isFull(){
 70:     return (size() == capacity - 1);
 71: }
 72:
 73: // ham tra ve gia tri phan tu o dau hang doi
 74: int ArrayQueue::front() throw (EmptyQueueException){
 75:     if(isEmpty()) throw EmptyQueueException();
 76:     return element[f];
 77: }
 78:
 79: // ham tra ve gia tri phan tu o cuoi hang doi
 80: int ArrayQueue::rear() throw (EmptyQueueException){
 81:     if(isEmpty()) throw EmptyQueueException();
 82:     return element[(capacity + r - 1) % capacity];
 83: }
 84:
 85: // ham in hang doi
 86: void ArrayQueue::print(){
 87:     cout << "(";
 88:     int s = size();
 89:     for(int i = f, j = 0; j < s - 1; i = (i + 1) % capacity, j++){
 90:         cout << element[i] << ", ";
 91:     }
 92:     if(s > 0) cout << rear();
 93:     cout << ")" << endl;
 94: }
 95:
 96: // ---------------------------------------------------------------------------
 97: // doan chuong trinh test
 98: int main(){
 99:     ArrayQueue queue1;
100:     queue1.print();
101:
102:
103:     fflush(stdin);
104:     cin.get();
105:     return 0;
106: }
107:
```