# include <iostream>

# define MAX 5

using namespace std;

template <typename p>

class queue {

private:

p array[MAX];

p first;

p last;

public:

queue () {

first = -1;

last = -1;

}

void enqueue (int ele) {

if(!isFull()) {

array[++last] = ele;

if (ifSame()) { // same

last = -1;

first = -1;

}

}

else {

array[0] = ele;

last = 0;

}

}

p deque () {

if(!isEmpty()) {

first++;

if (ifSame()) { // same

last = -1;

first = -1;

}

}

else {

cout << endl << "UNDERFLOW";

}

}

p returnFirstEle() {

return array[first];

}

p returnFirst () {

return first;

}

p returnLast () {

return last;

}

p returnEle(int val) {

return array[val];

}

bool ifSame () {

if (last == first) { // same

return true;

}

else {

return false;

}

}

bool isEmpty () {

if (first == -1 & last == -1)

return true;

else

return false;

}

bool isFull () {

if (first != -1 & last == MAX-1)

return true;

else

return false;

}

};

int main () {

queue<int> s1;

char chr;

do {

int ch = 0, ele = 0, first = 0, last = 0;

cout << endl << "> 1. Push onto queue";

cout << endl << "> 2. Pop from queue";

cout << endl << "> 3. Print the queue";

cout << endl << "> 4. QUIT";

cout << endl << "Enter your choice: ";

cin >> ch;

switch (ch) {

case 1 : cout << endl << "Enter element to put onto queue: ";

cin >> ele;

s1.enqueue(ele);

break;

case 2 : s1.deque();

ele = s1.returnFirstEle();

cout << endl << ele << " has been successfully delete from the queue!";

break;

case 3 : first = s1.returnFirst();

last = s1.returnLast();

cout << endl;

while (first <= last) {

cout << s1.returnEle(++first) << "\t";

}

cout << endl;

break;

case 4 : exit(0);

default : cout << "Unfortunately your input can not be processed any longer";

break;

}

cout << endl << "Do you want to enter more? (Y/N): ";

cin >> chr;

} while (chr == 'Y' | chr == 'y');

return 0;

}