**package** Task1;

**import** java.util.Stack;

**public** **class** StackQuestion {

**public** **static** **void** main(String[] args) {

// The test cases

Stack<Integer> stk = **new** Stack<Integer>();

stk.push(5);

stk.push(7);

stk.push(6);

stk.push(4);

stk.push(0);

System.***out***.println(*sortStack*(stk));

Stack<Integer> stk2 = **new** Stack<Integer>();

stk2.push(199);

stk2.push(200);

stk2.push(0);

stk2.push(549);

stk2.push(1099);

System.***out***.println(*sortStack*(stk2));

}

//**TODO** Complete the sortStack method to sort the input stack

**public** **static** Stack<Integer> sortStack(Stack<Integer> input) {

//Add your code here to sort Stacks such that

//the smallest items are on the top

**if** (input.size() <= 1) {

**return** input;

}

Stack<Integer> left = **new** Stack<Integer>();

Stack<Integer> right = **new** Stack<Integer>();

**int** count = 0;

**while** (input.size() != 0) {

count++;

**if** (count % 2 == 0) {

left.push(input.pop());

} **else** {

right.push(input.pop());

}

}

left = *sortStack*(left);

right = *sortStack*(right);

**while** (left.size() > 0 || right.size() > 0) {

**if** (left.size() == 0) {

input.push(right.pop());

} **else** **if** (right.size() == 0) {

input.push(left.pop());

} **else** **if** (right.peek().compareTo(left.peek()) <= 0) {

input.push(left.pop());

} **else** {

input.push(right.pop());

}

}

Stack<Integer> reverseStack = **new** Stack<Integer>();

**while** (input.size() > 0) {

reverseStack.push(input.pop());

}

**return** reverseStack;

}

}