```cpp
#include <iostream>
using namespace std;

//#include <time.h>
#include <string>
#include <sstream>

#include "Array.h"
#include "PriorityQueue.h"

//-----------------------------------------------------------
//for detecting memory leaks:
#define _CRTDBG_MAP_ALLOC
#include <crtdbg.h>
#ifdef _DEBUG
#ifndef DBG_NEW
#define DBG_NEW new ( _NORMAL_BLOCK , __FILE__ , __LINE__ )
#define new DBG_NEW
#endif
#endif
//-----------------------------------------------------------

void test()
{
    PriorityQueue<int> pq_int;
    for (int i = 0; i < 5; i++)
        pq_int.push(i);
    cout << "top = " << pq_int.top() << endl;
    cout << "int queue: " << endl;
    cout << pq_int;
    cout <<
      endl;
          //added
    cout << "and to make sure you didn't change the queue: " << endl;
    cout << pq_int;
    cout <<
      endl;
          //added
    //-----------------------------------------------------------
    PriorityQueue<string> pq_str;
    pq_str.push("This");
    pq_str.push("is");
    pq_str.push("project");
    pq_str.push("3");
    cout << "top = " << pq_str.top() << endl;
    cout << "str queue: " << endl;
    cout << pq_str;
    cout <<
      endl;
          //added
    //-----------------------------------------------------------
    Array<double> da1(5);
    for (int i = 0; i < 5; i++)
```

```
48          da1[i] = i;
49      Array<double> da2(2);
50      for (int i = 0; i < 2; i++)
51          da2[i] = i;
52      PriorityQueue<Array<double>> pq_array;
53      pq_array.push(da1);
54      pq_array.push(da2);
55      cout << "Array queue reversed: " << endl;
56      print_reversed_queue(pq_array);
57      cout <<                                                          ↵
          endl;                                                        // ↵
          added
58      //-------------------------------------------------------------
59      Array<PriorityQueue<int>> arr_pq_int;
60      arr_pq_int[0] = pq_int;
61      arr_pq_int[1] = pq_int;
62      cout << "Array of priority queues of ints: " << endl;
63      cout << arr_pq_int;
64      cout <<                                                          ↵
          endl;                                                    //      ↵
          added
65  }
66
67  int main(void)
68  {
69      test();
70      //memory leak:
71      cout << endl;
72      cout << "leaks: " << _CrtDumpMemoryLeaks();
73
74      return 0;
75  }
```