CSCI 3333 Homework MISS: The Case of the Missing Int(s)[1]

Due Saturday, September 22 at 5:00 PM

1 Introduction

In a distant land lies a curious town filled with strange inhabitants. They speak only in complementary language: everyone says everything *except* what they mean. When it comes to specifying numbers, like the number of dollars in their bank account, they don't say the number, but instead list every possible number (say, those between 1 and 1000000) *except* the actual value. For instance:

```
tell(1);   ("It's not 1")
                                  "Ok"
tell(2);   ("It's not 2")
                                  "Ok"
tell(4);   ("It's not 4")
                                  "Ok"
tell(5);   ("It's not 5")
                                  "Ok"

                                  ...

tell(999999);   ("It's not 999999")
                                  "Ok"
tell(1000000);   ("It's not 1000000")
                                  "Ok"
missing_one(x);   ("What number was missing?")
                                  x = 3;   ("3 was missing")
```

In this homework, you'll implement an "oracle" that can be told a sequence of numbers containing every number from 1 to 1000000, except for one or two, and returns the missing number(s).[2] If this seems impossible, just consider the following facts:

Theorem 1.1. *Let $x_1, x_2, \ldots, x_{n-1}, y$ be a permutation of the numbers from 1 to n. Then $y = \frac{n(n+1)}{2} - \sum_{i=1}^{n-1} x_i$.*

Theorem 1.2. *Let $x_1, x_2, \ldots, x_{n-2}, y_1, y_2$ be a permutation of the numbers from 1 to n. Let $s = y_1 + y_2$ and $t = y_1^2 + y_2^2$. Then $s = n(n+1)/2 - \sum_{i=1}^{n-2} x_i$ and $t = \frac{n(n+1)(2n+1)}{6} - \sum_{i=1}^{n-2} x_i^2$. Moreover, $2y_1^2 - 2sy_1 + s^2 - t = 0$.*

Theorem 1.3 (Quadratic formula)**.** *Let $a, b, c, x \in \mathbb{R}$ with $ax^2 + bx + c = 0$. Then $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.*

[1] This homework is inspired by Bill Gasarch's "Find the Missing Number or Numbers: An Exposition".

[2] In the example above, the oracle is the one responding "Ok" at each step, and "3 was missing" at the end.

2 Instructions

The following files have been given to you:

1. A C++ header file (`oracle.h`) declaring the Oracle class.
2. A C++ source file (`main.cpp`) containing a `main` function with tests.

Download the files at http://andrewwinslow.com/3333/hwMISS/. Create a new C++ source file named `oracle.cpp` that implements the class declared in `oracle.h`, so that `oracle.cpp` and the provided files compile into a program that runs with no failed tests. Submit the source file `oracle.cpp`.

3 Submission and Grading

Submit the aforementioned source file(s) via Blackboard as attached file(s). In the case of multiple submissions, the last submission before the deadline is graded.

For grading, each submission is compiled with the provided files and run. Submissions that do not run to completion (i.e. fail to print "Assignment complete.") receive no credit. Submissions that take an unreasonable amount of time (e.g. more than a minute or so) to run and do not meet the asymptotic efficiency requirements receive no credit. All other submissions receive full credit.

See the course late work policy for information about receiving partial credit for late submissions.

4 Tips

Here are some helpful tips for this assignment:

- Use `long long int`s everywhere to avoid overflow and remember that a constant like `1000000` is an `int`.
- Use the theorems in the assignment pdf. No mathematics is needed beyond these three theorems.
- Don't overthink the assignment: this assignment is more about algebra than algorithms.