Exercise 3

*Tasks marked with a * are assessed coursework. Hand in your solutions to these via email to* `rn@ic.ac.uk`. *(Resit students do not need to submit coursework.) Use the subject line "*`C++ CW: surname_firstname_CW3`*", where* `surname_firstname_CW3.cpp` *is the attached file that contains your solution. The course will be assessed based on 5 pieces of coursework (25%) and an end of term driving test (75%). Your submission must be* ***your own work*** *(submissions will be checked for plagiarism), and it should compile (and run) with the GNU C++ compiler* `g++`. *The deadline for submitting the coursework is 10pm on* **24/02/2019**.

---

1. *Extra long int*
   Create a class `extralongint` that can handle very long integer numbers, say with at least up to 10^8 digits. Code the mathematical operations `+`, `-`, `*` and `/`, for this class and provide ways to input and output its objects. Furthermore, overload the relational operators `<`, `>` and `==`. Let $a = 1234567890987654321$, $b = 9876543210123456789$. Compute the following values

   (a) $a * b$
   (b) $a * b/(b - a)$
   (c) $b^2 - a^2$
   (d) 51!
   (e) Is $21^{22} > 22^{21}$?

2*. *Fractions*
   Design a class `fraction` that allows you to store numbers in fractional form and do basic arithmetics with them. Your class declaration should include at least the following methods.

```
class fraction {
friend ostream &operator<< (ostream &os, const fraction &f);
private:
  int numerator, denominator;
  void reduce();
public:
  fraction(int n = 0, int d = 1) : numerator(n), denominator(d) { reduce(); }
  fraction operator+ (const fraction &f) const;
  fraction &operator+= (const fraction &f);
  bool operator< (const fraction &f) const;
  fraction operator- () const { fraction res(-numerator, denominator); return res; }
};
```

   In particular, your class should execute all of the following statements correctly. Any fraction $f = \frac{p}{q} \in \mathbb{Q}$ with $p \in \mathbb{Z}$, $q \in \mathbb{N}$, should be printed in its reduced form as '`p/q`', while fractions $f = i \in \mathbb{Z}$ should be printed as '`i`'.

```
int main() {
  fraction a(1,12), b(1,25), c(1,300), e(3,4), f(2,5), g(7,2), h;
  cout << a << " + " << b << " + " << c << " = " << a+b+c << endl;
  if (-a < b) cout << -a << " < " << b << endl;
  else cout << -a << " >= " << b << endl;
  h = e*f-g;
  cout << e << " * " << f << " - " << g << " = " << h << endl;
  fraction i(-1,6), j(5,17), k(3,5);
  cout << i << " + " << j << " / " << k << " = " << i+j/k << endl;
  int m = 2, q = 1;
  fraction n(1,3), p(4,5), s(a), t(b), u(c), w(e), x(f), y(g), v, z;
  cout << m << " + " << n << " = " << m+n << endl;
  cout << p << " - " << q << " = " << p-q << endl;
  v = (a+=b-=u);
  cout << s << " += " << t << " -= " << u << " = " << v << endl;
  z = (e*=f/=g);
  cout << w << " *= " << x << " /= " << y << " = " << z << endl;
  double d = (double) e;
  if (d < f) cout << d << " < " << f << endl;
  else cout << d << " >= " << f << endl;
  return 0;
}
```