Homework Assignment #6

Due Time/Date

2:20PM Wednesday, December 11, 2024. Late submission will be penalized by 20% for each working day overdue.

How to Submit

This assignment must be carried out using Coq and Frama-C. Please email your completed homework in one single .zip file to the instructor by the due time. You may discuss the problems with others, but copying answers is strictly forbidden.

Problems

1. (30 points) Prove the following lemmas using Coq. The predicate `Zis_gcd` in Lemma `gcd_equiv` is defined in the `ZArith.Znumtheory` library such that `Zis_gcd a b d` asserts that `d` is the GCD of `a` and `b`. (Hint: for constructing the proofs, most needed lemmas may be found in the `ZArith.BinInt` library.)

```
Require Import ZArith.
Require Import ZArith.Znumtheory.

Open Scope Z_scope.

Definition isGCD(a b d: Z): Prop :=
  (d | a) /\ (d | b) /\
  (forall c: Z, (c | a) -> (c | b) -> (c | d)).

Lemma gcd_equiv: forall a b d: Z, Zis_gcd a b d <-> isGCD a b d.
Lemma div_minus_l: forall a b c: Z, (b > c) -> (a | (b - c)) ->
                   (a | c) -> (a | b).
Lemma div_minus_r: forall a b c: Z, (a | b) -> (a | c) ->
                   (b > c) -> (a | (b - c)).
Lemma gcd_refl: forall a: Z, isGCD a a a.
Lemma gcd_minus: forall a b d: Z, isGCD a b d ->
                 (a > b) -> isGCD (a-b) b d.
```

2. (30 points) Annotate the following C function to show that it preserves sortedness of the input arrays (i.e., the input arrays remain sorted if they were sorted), assuming no overflow will occur, and prove correctness of your annotation using Frama-C.

```
void addArray(int* a, int *b, int n)
{ int i;

  for (i=0; i<n; i++)
    a[i] = a[i] + b[i];
}
```

3. (40 points) Annotate the following C function to show its behavior and prove correctness of your annotation using Frama-C.

```
int originalEuclid(int m, int n)
{ int x,y,tmp;

  x = m;
  y = n;
  while (x != y) {
    if (x < y) {
      tmp = x;
      x = y;
      y = tmp;
    }
    x = x - y;
  }
  return x;
}
```