LAB02 Preparation (P16-P20)

All problems (P16-P20) use integer arithmetic (set $\mathbb{Z}$). They should be enclosed in a Z-scope. And some imports and definitions are also assumed:

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

Open Scope Z_scope.

Lemma sample2_16 ...
Lemma sample2_17 ...
Lemma sample2_18 ...
Lemma sample2_19 ...
Lemma sample2_20 ...

Close Scope Z_scope
```

Reusing Existing Work

In many cases you have an option – prove simple lemmas on your own, or find them to reuse.
Here is a useful lemma: number 1 divides any other integer number $a \in \mathbb{Z}$.

```
Lemma sample2_16_helper:
    forall a: Z, (1 | a).
Proof.
  intros a.
  unfold Z.divide.
  exists a.
  ring.
Qed.
```

If you want to find lemmas containing subexpresison $(1 \mid \ldots)$, you can run search:

```
Locate "|".
```

This should return this answer:

```
"( x | y )" := Z.divide x y : Z_scope
        (default interpretation)
```

This immediately suggests that you search for `Z.divide`:

```
Search Z.divide.
```

It will print out all the lemmas containing divisibility relation on integers (the set $\mathbb{Z}$). Beware of similar lemmas in other number sets (e.g. "nat" - nonnegative integers). It is a different set, and these theorems are not applicable for your problems.

```
Z.divide_1_l: forall n : Z, (1 | n)
```

This lemma shows that each integer number is divisible by 1. In our case it was easy to prove it independently, but usually searching for an existing result saves your time.

Look Around Before you Prove

(A) These will print out specific lemmas or definitions:

```
Print rel_prime.
Print Z.add_simpl_l.
Print Z.mul_add_distr_r.
Print Z.pow_pos.
```

(B) Search all lemmas containing some concept:

```
Search Zis_gcd.
Search Z.divide.
Search Z.divide.
```

(C) Search algebraic patterns with wildcards. The first two patterns might display some lemmas how to open parentheses using distributivity.

```
SearchRewrite (_ * (_ + _)).
SearchRewrite ((_ + _) * _).
SearchRewrite (_ * (_ * _)).
SearchRewrite (_ + _ - _).
SearchRewrite (_ - _).
SearchRewrite (_ + _).
SearchRewrite (_ * _).
SearchRewrite (1 * _).
```

If you run `SearchRewrite` for a single "minus":

```
Zminus_0_l_reverse:
    forall n : Z, n = n - 0
Zminus_diag_reverse:
    forall n : Z, 0 = n - n
Zplus_minus_eq:
    forall n m p : Z, n = m + p -> p = n - m
Zminus_plus_simpl_l:
    forall n m p : Z, p + n - (p + m) = n - m
Zminus_plus_simpl_l_reverse:
    forall n m p : Z, n - m = p + n - (p + m)
Zminus_plus_simpl_r:
    forall n m p : Z, n + p - (m + p) = n - m
Zeq_minus: forall n m : Z,
    n = m -> n - m = 0
Lemma rel_prime_bezout :
  forall a b:Z, rel_prime a b -> Bezout a b 1.
```

(D) Locate some notations:

```
Locate "|".
Locate "^".
```

Problem 16. If $\gcd(a,b) = 1$ and c divides a, then $\gcd(b,c) = 1$.

Hints: Use the definition of GCD (greatest common divisor) in your hypothesis: We know that every k that is a divisor of both a and b is equal to 1. Now assume that there is some number (possibly a different k') that divides both b and c. You must show that it also must be equal to 1.
You may need to prove or find the following lemma: $\forall a, b, c \in \mathbb{Z},\ (a \mid b) \to (b \mid c) \to (a \mid c)$. Namely, the divisibility relation is *transitive*: If a divides b and b divides c, then a divides c.
If you see `Zis_gcd` as one of the hypotheses, you can destruct the hypothesis:

```
destruct H as [_ _ H1].
```

Here the underscores are placeholders (they tell that you will not use these hypotheses, so you are not giving them any names).
If you see `Zis_gcd` in your goal, then type this tactic:

```
apply Zis_gcd_intro.
```

It will create a different goal - one that actually describes the meaning of the greatest common divisor.

Problem 17. If $\gcd(a, b) = 1$, then $\gcd(ac, b) = \gcd(c, b)$.
Hints: This theorem is easiest to solve using Bezout's indentity. You may need a warm-up – several lemmas:

(A) $\forall n \in \mathbb{Z},\ 1 \cdot n = n$.

(B) $\forall a, b, c \in \mathbb{Z},\ (a \mid b) \to (a \mid c \cdot b)$.

(C) $\forall a, b, c, k \in \mathbb{Z},\ \gcd(a, b) = 1 \to (k \mid a \cdot c) \to (k \mid b) \to (k \mid c)$.

The first two lemmas are very easy. The last lemma may need to use Bezout's identity: Once a and b are mutual primes, there should be integers x, y such that $ax + by = 1$. This can be used to prove that k must divide c (if we already know that k divides $a \cdot c$ and b. In order to prove the last lemma (and also the Problem 17 itself) you may need to use several lemmas (that already exist in Coq). Please take a look at these:

```
rel_prime_bezout
mul_1_left
Z.mul_add_distr
Z.mul_assoc
div_multiple_left
```

Problem 18. If $\gcd(a, b) = 1$ and c divides $(a+b)$, then $\gcd(a, c) = \gcd(b, c) = 1$.
Hints. This theorem may need a few easy lemmas:

(A) If $a = b$ and $c = d$ then $a - c = b - d$.
(B) If $\gcd(a, b) = 1$ and c divides $a + b$, then $\gcd(a, c) = 1$.

A few more predefined lemmas may be useful. For example,

```
Z.divide_1_l
Z.add_comm
Zis_gcd_sym
```

Problem 19. If $\gcd(a, b) = 1$; d divides ac; d divides bc, then d divides c.
Hints. You may need to state the Bezout's identity, then "destruct" it – find those specific x, y which satisfy $ax + by = 1$. Then you can find out about divisors of c as well.

Problem 20. If $\gcd(a, b) = 1$, then $\gcd(a^2, b^2) = 1$.
Hints. Bezout's identity may work. But you may also want to prove a chain of equalities: $\gcd(a, b) = \gcd(a, b^2) = \gcd(a^2, b^2)$. In order to prove this, you need to use the fact that a,b are mutually prime.

Problem 15. In this task you need to define a function (using recursive "fixpoint") to compute the result of some procedure where we remove numbers from a list. See https://bit.ly/2VhNikC for details.