Извлечение кода: алгоритм Евклида

1. Докажите вспомогательные леммы:

```
Definition div (x : nat) (y : nat) := exists z : nat, x = y * z.

Lemma EA1 : forall b q r x : nat,
(div b x -> div r x -> div (q * b + r) x).

Lemma EA2 : forall b q r x : nat,
(div b x -> div (q * b + r) x -> div r x).
```

2. Докажите (постройте терм) следующий принцип рекурсии по двум натуральным аргументам:

```
Theorem lt_wf_duple_rec : forall n m : nat, forall P : nat -> nat -> Set,
(forall n0 : nat, forall m0 : nat, (forall n1 : nat, n1 < n0 -> P n1 m0) ->
(forall m1 : nat, m1 < m0 -> P n0 m1) -> P n0 m0) -> P n m.
```

Подсказка: для доказательства можно воспользоваться вспомогательным принципом рекурсии по сумме аргументов:

```
Theorem lt_wf_sum_rec : forall n m : nat, forall P : nat -> nat -> Set,
(forall n0 m0 : nat, (forall n1 m1 : nat, n1 + m1 < n0 + m0 -> P n1 m1) -> P n0 m0)
-> P n m.
```

3. Постройте терм следующего типа:

```
Definition GCD : forall a b : nat,
{ d : nat | div a d /\ div b d /\ forall d2 : nat, (div a d2 /\ div b d2 -> div d d2) }.
```

и извлеките программный код (на языке OCaml), вычисляющий наибольший общий делитель двух натуральных чисел.