```
theory Problem-1
  imports Main
begin

theorem problem1:
  fixes f :: int ⇒ int
  obtains k where
    (∀ a b. f (2*a) + 2*f b = f (f (a + b))) ⟷
      (∀ x. f x = 2*x + k) ∨ (∀ x. f x = 0)
proof (rule, rule)
  assume ∀ a b. f (2*a) + 2*f b = f (f (a + b))
  then have eq: f (2*a) + 2*f b = f (f (a + b)) for a b by auto
  have f (2*a) + 2*f b = f (2*b) + 2*f a for a b
    using eq[of a b] and eq[of b a]
    by (simp add: add.commute)
  from this[of 0] have [simp]: f (2*a) = 2*f a − f 0 for a by simp
  have eq': 2*f a + 2*f b − f 0 = f (f (a + b)) for a b
    using eq[of a b] by simp
  have 2*f a + f 0 = f (f a) for a
    using eq'[of a 0] by simp
  hence [simp]: f (f a) = 2*f a + f 0 for a..
  from eq' have 2*f a + 2*f b − f 0 = 2*f (a+b) + f 0 for a b by simp
  hence 2*f a + 2*f b − 2*f 0 = 2*f (a + b) for a b by (simp add: ac-simps)
  hence eq'': f a + f b − f 0 = f (a + b) for a b by smt

  define m c where
    m = f 1 − f 0 and
    c = f 0
  have nat-linear: f (int n) = m*(int n) + c for n :: nat
  proof (induction n)
    case 0
    then show ?case unfolding m-def c-def by simp
  next
    case (Suc n)
    then show ?case
      unfolding m-def c-def
      by (simp flip: eq''[of 1 int n] add: ac-simps distrib-right)
  qed

  have f-neg: f (−a) = 2*f 0 − f a for a
    using eq''[of a −a] by simp

  have linear: f x = m*x + c for x
  proof (cases x ≥ 0)
    case True
    then show ?thesis
      using nat-linear[of nat x] by simp
  next
    case False
    then show ?thesis
      using nat-linear[of nat (−x)] f-neg by (simp add: c-def)
  qed

  hence params: 2*m*(a+b) + 3*c = m*m*(a+b)+m*c+c for a b :: int
    using eq[of a b] by (simp add: algebra-simps)

  from params[of 0 0] and params[of 1 0] have 2*m = m*m by algebra
  then consider m = 2 | m = 0 by auto
  then show (∀ x. f x = 2*x + c) ∨ (∀ x. f x = 0)
  proof cases
    case 1
    then have f x = 2*x + c for x
      using linear by simp
```

```
    then show ?thesis by simp
  next
    case 2
    with params[of 0 0] have c = 0 by simp
    with linear and ‹m = 0› have f x = 0 for x by simp
    then show ?thesis by simp
  qed
next
  define c where c = f 0
  assume (∀ x. f x = 2*x + c) ∨ (∀ x. f x = 0)
  then show (∀ a b. f (2*a) + 2*f b = f (f (a + b)))
    by auto
qed

end
```