Inductive block_description_from_to (m : mem) ($heap$: block) : int → int → list int → **Prop** :=
| block_description_end : ∀ he, block_description_from_to m $heap$ he he nil
| block_description_block : ∀ he n, ($n = he \rightarrow$ **False**) →
∀ v, Some (Vint v) = load Mint32 m $heap$ (signed n) →
(Size_header $v \neq$ zero → ∀ x, modulo_four x n → cmp Cle (add n four) x = true →
cmp Cle x (add n (Size_header v)) = true →
valid_access m Mint32 $heap$ (signed x)) →
cmp Cle n (add n three) = true →
cmp Cle (add n three) (add (add n (Size_header v)) three) = true →
cmp Cle (add (add n (Size_header v)) three) (sub he one) = true →
∀ l, block_description_from_to m $heap$ (add (add n (Size_header v)) four) he →
block_description_from_to m $heap$ n he (add n four :: l).