Name: **Noman Siddique**

Roll Number: **P191664**

Assignment: **03**

Course: **Formal Method in software Engineering**

Instructor: **Shakir Ullah**

Statement:

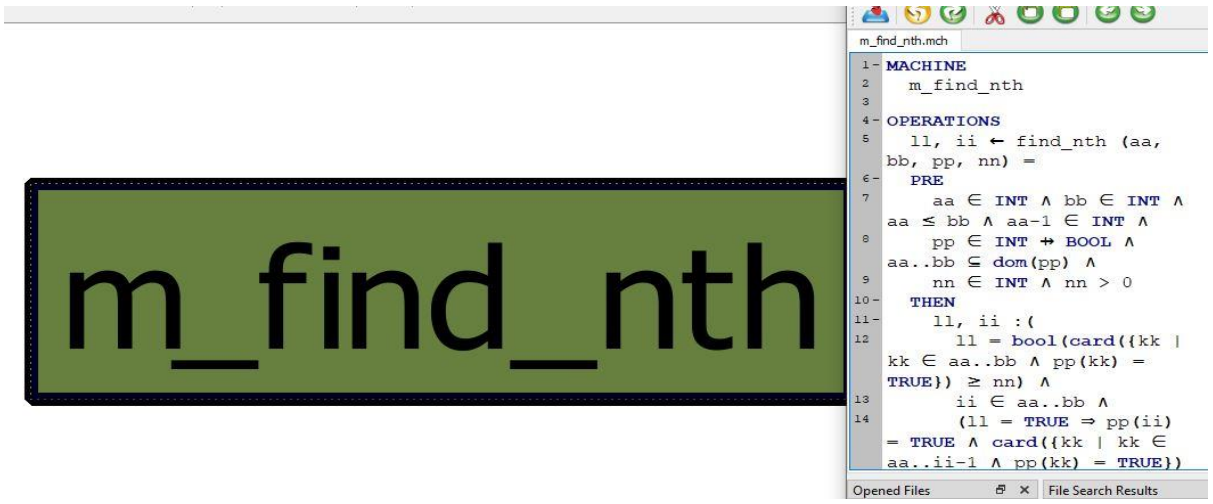
The problem that we're going to solve is finding elements of a sequence satisfying a given predicate. More precisely, we're interested in finding exactly the n^{th} such element, for a given value of *n*.

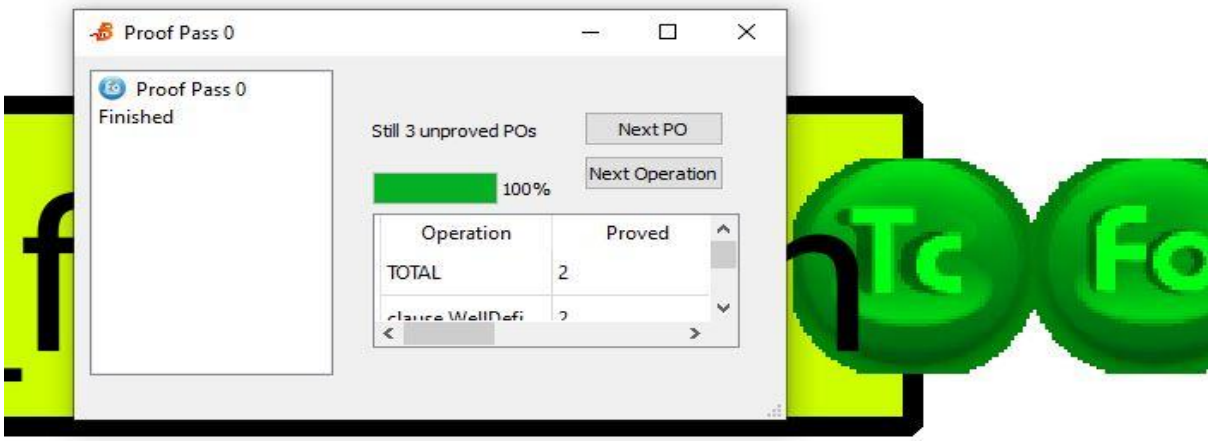
Solution:

```
MACHINE
m_find_nth

OPERATIONS
ll, ii ← find_nth (aa, bb, pp, nn) =
PRE
aa ∈ INT ∧ bb ∈ INT ∧ aa ≤ bb ∧ aa-1 ∈ INT ∧
pp ∈ INT ⇸ BOOL ∧ aa..bb ⊆ dom(pp) ∧
nn ∈ INT ∧ nn > 0
THEN
ll, ii :(
ll = bool(card({kk | kk ∈ aa..bb ∧ pp(kk) = TRUE}) ≥ nn) ∧
ii ∈ aa..bb ∧
(ll = TRUE ⇒ pp(ii) = TRUE ∧ card({kk | kk ∈ aa..ii-1 ∧ pp(kk)
= TRUE}) = nn-1))
END
END
```

OUTPUT:





CODE:

```
MACHINE
Access
SETS USER; PRINTER; OPTION; PERMISSION = { ok, noaccess }
CONSTANTS options
PROPERTIES
options : PRINTER <-> OPTION &
dom( options ) = PRINTER & ran( options ) = OPTION
VARIABLES access
INVARIANT access : USER <-> PRINTER
INITIALISATION access := {}
OPERATIONS
add (uu, pp) =
PRE uu:USER & pp:PRINTER
THEN access := access \/ { uu |-> pp }
END ;


END
```

OUTPUT :


Access