ESERCIZI CON ECLIPSE

Ho un array di 10 luci ognuna della quale può essere in stato acceso o spento.

Inizialmente sono tutte accese.

Posso cambiare lo stato di una lampadina **singola** (toggle).

Posso decidere si spegnerle tutte (allOff) e questa operazione ha priorità rispetto al cambio di stato di una lampadina.

Nota che potrei anche non dare alcun comando di cambio e lasciare le lampadine come sono.

5. Asmeta

Scrivi la specifica asmeta (usa più costrutti possibili, ad esempio derivate, macro rule etc)

Scrivi le seguenti proprietà e dimostrale (o prova che sono false)

- è sempre possibile avere tutte le lampadine spente
- Se sono tutte spente, fino a quando non do il comando di toggle su una lampada, rimangono tutte spente
- Se do il comando ad una lampadina, nello stato successivo essa cambia stato

Scrivi inoltre almeno una proprietà che è giustamente falsa e controlla che il model checker trovi il contro esempio atteso (puoi partire anche dalle proprietà sopra).

Scrivi almeno un'altra proprietà di safety e una liveness che siano vere.

Scrivi uno scenario avalla di un caso significativo (ad esempio spengo tutte le lampadine).

6. JML

Scrivi il codice in Java con in contratti opportuni (la classe LightArray ha uno costruttore, il metodo alloff che spenge tutte le lampadine e toggle(i) che cambia lo stato alla lampadina i-esima).

Cerca di scrivere sia le precondizioni, che le postcondizioni dei metodi.

Cerca di scrivere anche invarianti.

Testa i contratti JML con una classe main in cui chiami i diversi metodi.

Prova anche a modificare il codice e controlla che i contratti siano violati. Documenta bene le violazioni e le loro cause in commenti e nel file di documento.

6. KEY

Copia il progetto precedente, togli tutte le cose statiche (anche il main) non usare gli enumerativi. Dimostra con key che i contratti siano rispettati.

Target	Contract	Proof Reuse	Proof Result
LightArray()	JML operation contract 0	New Proof	Closed
alloff()	JML operation contract 0	New Proof	Closed
toggle(int)	JML operation contract 0	New Proof	Closed

Per dire che una lampadina è accesa/spenta ho usato i booleani. Ho usato diverges true e anche invaranti oltre a loop invariant

7. MODEL BASED TESTING

Traduci lo scenario avalla nel caso di test JUNIT per il codice java che hai scritto al punto 5.