

# LOGICA PER LA PROGRAMMAZIONE – a.a. 2016/17

## Sesta Esercitazione — 15/12/2016

**ESERCIZIO 1** Si consideri il seguente frammento di programma annotato

```
{x = 0 ∧ y = 0 ∧ n ≥ 0}
{Inv : y = x * n ∧ x ∈ [0, n]}{t: n - x}
while (x < n) do
  x, y := x + 1, y + n;
endw
{y = n2}
```

1. Si scrivano le ipotesi di progresso, di invarianza e di terminazione.
2. Si dimostrino le ipotesi di progresso e di invarianza.

**ESERCIZIO 2** Si verifichi la seguente tripla di Hoare (assumendo **a**: array [0, n) of nat)

```
{P}
if a[x] > a[x - 1] then sum := sum + 2 * a[x] else skip fi
{Q}
```

dove

- $P \equiv x \in [1, n) \wedge sum = (\sum i : i \in [1, x) \wedge a[i] > a[i - 1] \cdot 2 * a[i])$ ,
- $Q \equiv sum = (\sum i : i \in [1, x) \wedge a[i] > a[i - 1] \cdot 2 * a[i])$ .

**ESERCIZIO 3** Si verifichi la seguente tripla di Hoare (assumendo **a**: array [0, n) of nat)

```
{h ∈ dom(a) ∧ h ≥ 1 ∧ (∀i. i ∈ [0, h) ⇒ a[i] > k)}
a[h] := a[0] + 1
{(∀i. i ∈ [0, h) ⇒ a[i] > k)}
```

**ESERCIZIO 4** Si consideri il seguente frammento di programma annotato

```
{x = 0 ∧ z = 1 ∧ n ≥ 0 ∧ m ≥ 0}
{Inv : x ∈ [0, max(n, m)] ∧ z = wx}{t: max(n, m) - x}
while (x < n or x < m) do
  z := z * w;
  x := x + 1
endw
{z = wmax(m, n)}}
```

1. Si scrivano le ipotesi di invarianza, di progresso e di terminazione.
2. Si dimostri l'ipotesi di invarianza.