

Ingegneria del Software – a.a. 2009/10

Appello del 23 Luglio 2010

Cognome

Nome

Matricola

Istruzioni

1. La mancata indicazione dei dati anagrafici e della sezione comporta l'annullamento del compito.
2. Al termine, consegnare **solo i fogli distribuiti** utilizzando il **retro** delle pagine in caso di necessità. Non separare questi fogli. Eventuali fogli di brutta, ecc. **non** verranno **in nessun caso** presi in considerazione. È possibile scrivere **in matita**.
3. È possibile consultare liberamente libri, manuali o appunti. È **proibito** l'uso di ogni dispositivo elettronico (quali calcolatrici tascabili, telefoni cellulari, ecc.).
4. Non è possibile lasciare l'aula conservando il tema della prova in corso.
5. Tempo a disposizione: 2h.
6. Punteggio totale a disposizione: 100 punti nominali. La sufficienza si raggiunge indicativamente con 60 punti.

Esercizio 1: .....

Esercizio 2: .....

Esercizio 3: .....

Totale .....

### Esercizio 1 (punti 25)

E' data l'astrazione procedurale **calcolo**. Il metodo riceve come parametro un array x di almeno 2 interi e restituisce un array di interi, i cui elementi sono così costituiti: il primo elemento è la somma di tutti gli elementi di x, il secondo è la somma dei primi x.length - 1 elementi di x, il terzo dei primi x.length - 2 ecc.

a- Si scrivano la pre- e post-condizione del metodo evidenziando tutti gli elementi significativi

**SOL:**

```
//@ requires (x != null) && (x.length >= 2)
```

```
/*@ ensures \result != null && \result.length == x.length &&
```

```
@ (\forall int i; i >= 0 && i < x.length; (\result[i]
```

```
@ == (\sum int j; j >= 0 && j < (x.length - i); x[j])))
```

```
@*/
```

```
static int[] calcolo(int[] x)
```

b- Si identifichino le eccezioni necessarie per rendere totale il metodo e si scrivano le opportune modifiche alla specifica di cui al punto (a).

**SOL:**

```
//@ requires true
```

```
/*@ ensures ((x != null) && (x.length >= 2)) =>
```

```
@ \result != null && \result.length == x.length &&
```

```
@ (\forall int i; i >= 0 && i < x.length; \result[i]
```

```
@ == (\sum int j; j >= 0 && j < (x.length - i); x[j])))
```

```
@*/ // @ signals (NullPointerException e) (x == null)
```

```
// @ signals (ArrayTooSmallException e) ((x != null) && (x.length < 2))
```