# PROGRAMMAZIONE II (A,B) - a.a. 2013-14
# Quinto Appello - Prova Scritta del 13 Febbraio 2015

**Esercizio 1.** Si consideri il seguente programma OCaml:

```
let x = 5;;
let z = 4;;
let f = fun y ->  if x = 1 then let k w = w+y+x in k
                           else let g w = w+y+z in g;;
let h = f 3;;
h 2;;
```

1. Indicare il tipo inferito dall'interprete OCaml per la funzione `f`.
2. Simulare la valutazione del programma mostrando la struttura della pila dei record di attivazione.
3. Indicare il valore restituito dal programma.

**Esercizio 2.** Si consideri il tipo di dati astratto modificabile `IntegerCollection`, utilizzato per rappresentare una collezione mutabile di oggetti di tipo `Integer`. Il tipo di dati astratto `IntegerCollection` ha, tra gli altri, i seguenti metodi:

- `public int occurrences (Integer elem)`

  che restituisce il numero di occorrenze del parametro `elem` nella collezione;
- `public int size ()`

  che restituisce la cardinalità della collezione (il numero degli elementi presenti);
- `public void insert (Integer elem)`

  che inserisce l'oggetto `elem` nella collezione;
- `public Integer extractMax ()`

  che restituisce ed elimina dalla collezione una occorrenza dell'elemento avente valore massimo.

1. Completare la specifica del tipo di dati astratto `IntegerCollection` (overview con descrizione di un'istanza tipica e specifica completa dei metodi, compreso eventuali eccezioni lanciate).
2. Definire una implementazione del tipo di dati astratto `IntegerCollection`. L'implementazione deve utilizzare come struttura di supporto un `Vector` che contiene oggetti di tipo `Integer`:
   - `Vector <Integer> collection` .
3. Definire l'invariante di rappresentazione e dimostrare che l'implementazione del metodo `extractMax` preserva tale invariante.
4. Si consideri il tipo di dati astratto `IntegerCollection_1`, identico al precedente tranne che per il metodo `extractMax`, il quale elimina dalla collezione tutte le occorrenze dell'elemento di valore massimo, restituendone una. Fornire la specifica completa del metodo `extractMax` di `IntegerCollection_1`. Mostrare che il tipo di dati astratto `IntegerCollection_1` non è un sottotipo di `IntegerCollection`, spiegando quali regole del Principio di Sostituzione sono violate.

**Esercizio 3.** Si consideri il seguente programma Java.

```
class A{
    int a=10;
    public void show(){
        System.out.println("Show A: "+a);
    }
}
class B extends A{
    public int b=20;
    public void show(){
        System.out.println("Show B: "+b);
    }
}
public class Example {
    public static void main(String[] args) {
        A aObj = new A();
        aObj.show();    // Punto (1)
        B bObj = new B();
        bObj.show();   //Punto (2)
        aObj = bObj;
        aObj.show();  //Punto(3)
        aObj = new B();
        aObj.show();  //Punto (4)
        System.out.println(bObj.b);  //Punto (5)
    }
}
```

Indicare il risultato dell'esecuzione dell'istruzione `println` nei punti (1)–(5) del `main`. Si motivi la risposta.