

# Esame di Programmazione II

Appello di giorno 23 Giugno 2017 - prova B  
Università degli Studi di Catania - Corso di Laurea in Informatica

## Testo della Prova

### *Definizione Iniziale.*

Ad un esame universitario un studente viene identificato dal suo *numero di matricola*. Dopo aver sostenuto l'esame, per stilare la valutazione finale, il professore tiene in considerazione il voto ottenuto, il numero di domande a cui lo studente ha risposto correttamente e, infine, l'età accademica dello studente, ovvero da quanti anni lo studente è iscritto all'università.

Dato lo studente  $S_i$ , indichiamo con  $v_i$  il suo voto, con  $r_i$  il numero di domande a cui il candidato ha risposto correttamente e con  $e_i$  l'età accademica dello studente.

In base alle direttive indicate dal docente uno studente  $S_i$  ottiene una valutazione superiore a quella dello studente  $S_j$  (quindi  $S_i < S_j$ ) se:

- $v_i > v_j$ ;
- $v_i = v_j$  e  $r_i > r_j$ ;
- $v_i = v_j$ ,  $r_i = r_j$  e  $e_i < e_j$ .

### 1. Classe **Studente** (10 punti)

Si implementi in C++ la classe **Studente** fornendo, oltre al costruttore, dei metodi per la modifica e la valutazione dei valori  $v$  (voto),  $r$  (numero di risposte corrette) ed  $e$  (età accademica). Si effettui inoltre l'overloading degli operatori di confronto ( $<$ ,  $>$ ,  $=$ ,  $!=$ ,  $<=$ ,  $>=$ ) in modo da poter confrontare istanze di questa classe.

### 2. Classe **BST** (12 punti)

Si implementi in C++ la classe **BST** rappresentante un insieme di elementi implementato mediante un albero binario di ricerca. Implementare l'albero utilizzando i Template. La classe deve permettere le seguenti operazioni:

- la ricerca di un elemento dell'insieme sulla base della sua chiave;
- l'inserimento di un nuovo elemento;
- la stampa degli elementi presenti nell'insieme.

### 3. Classe **Valutazione** (8 punti)

Si implementi in C++ la classe **Valutazione** rappresentante l'insieme degli studenti che hanno preso parte all'esame, ordinati in base alla loro valutazione complessiva. L'insieme deve essere implementato mediante un albero binario di ricerca, in base ai criteri forniti in precedenza. La classe deve permettere le seguenti operazioni:

- l'inserimento di un nuovo studente alla valutazione;
- la ricerca di uno studente sulla base della sua posizione nella valutazione;
- la stampa della valutazione di tutti gli studenti che hanno preso parte all'esame, in ordine non decrescente rispetto alla loro valutazione complessiva.

### Test e valutazione del proprio elaborato.

Per la verifica e la valutazione del proprio elaborato si eseguano le seguenti istruzioni:

```
// esercizio n.1
int v=89, r=60, e=27;
Studente *a = new Studente("X81000123", v, r, e);
Studente *b = new Studente("X81000452", 63, 63, 24);
Studente *c = new Studente("X81000104", 89, 60, 25);
if( *a < *b) cout << "True"; else cout << "False";
if( *a >= *b) cout << "True"; else cout << "False";
if( *a == *c) cout << "True"; else cout << "False";
if( *a <= *c) cout << "True"; else cout << "False";

// esercizio n.2
BST<int> *t = new BST<int>();
t->ins(2)->ins(3)->ins(12)->ins(4)->ins(9)->ins(5);
t->print();

// esercizio n.3
Valutazione *v = new Valutazione();
v->ins(*a)->ins(*b)->ins(*c);
cout << v->search(2);
v->print();
```

L'output del programma sarà quindi:

```
True
False
False
False
2, 3, 4, 5, 9, 12
X81000123
X81000104, X81000123, X81000452
```