

**Università di Napoli Federico II**  
**Corso di Laurea in Ingegneria Informatica**  
**Corso di Laurea in Ingegneria dell'Automazione**

**- Esame di Programmazione 1-**  
**Prova pratica del 3/02/2010**  
**Durata della prova: 2h**

**Cognome** ..... **Nome** ..... **Matr.** .....

**Testo della prova**

Si realizzino in linguaggio C++ le classi Fotografia e CodaFotografie di seguito descritte.

La classe Fotografia ha le variabili membro Titolo e Formato, entrambe di tipo char \*, ed implementa le funzionalità necessarie alla corretta gestione della estensione dinamica e le seguenti ulteriori funzioni membro:

- funzioni di accesso e posizionamento (get e set)
- overloading dell'operatore << sia per la scrittura su video che per la scrittura su file binario
- overloading dell'operatore >> per la lettura da file binario

Si supponga che le informazioni vengano scritte sul file binario memorizzando per ciascuna fotografia nel seguente ordine:

il numero di caratteri della stringa Titolo

la stringa Titolo (escluso il terminatore)

il numero di caratteri della stringa Formato

la stringa Formato (escluso il terminatore)

L'operatore >> opera sulla base del precedente formato (ricordarsi che sul file non è memorizzato il terminatore!)

*Si supponga inoltre che il file sia aperto e chiuso all'esterno degli operatori di flusso (nel chiamante).*

La classe CodaFotografie è realizzata mediante una *coda statica circolare*. Fornisce le consuete funzionalità (empty, full, pop, append, stampa) ma aggiunge un costruttore che riceve in ingresso il nome di un file ed inizializza la coda leggendo da file (a tale scopo utilizza l'opportuno operatore definito in Fotografia). *Per semplicità si supponga che il costruttore sappia quanti oggetti di tipo Fotografia sono stati "serializzati" sul file.*

Si generi infine una eccezione nel caso in cui si tenti di eliminare un elemento dalla coda vuota.

Si implementi un programma utente (main) di prova strutturato in due parti:

*la prima parte effettua il test della classe fotografia e genera un file utilizzato nella seconda parte che effettua il test della classe coda.* Non si dimentichi di testare la gestione dell'eccezione di coda vuota.

Si strutturi l'applicazione secondo i consueti principi di programmazione modulare.