

## Compito di Fondamenti di Informatica

(programma C++)

2 settembre 2006

1. [4 punti] Con riferimento al linguaggio C++, cosa si intende per memoria 'stack' (di run time) e memoria 'heap'? Come vengono allocate e rilasciate tali memorie?
2. [4 punti] Con riferimento all'ordinamento dei valori contenuti in un array di interi, descrivere l'idea alla base dell'algoritmo di ordinamento per selezione e discutere le prestazioni di tale algoritmo. Quali sono le prestazioni dell'algoritmo nel caso gli interi nell'array siano già ordinati?
3. [4 punti] Con riferimento al linguaggio C++, descrivere i meccanismi di chiamata per valore e chiamata per riferimento. In quali casi risulta conveniente utilizzare il secondo piuttosto che il primo?
4. [7 punti] Utilizzando la **ricorsione**, specificare una funzione C++ avente il prototipo `boolean trova(const int* array, int dim, int n)`. Tale funzione restituisce il valore `true` se l'elemento `n` è presente nell'array, ed il valore `false` in caso contrario. **Non** devono essere utilizzate variabili statiche o variabili globali.
5. [16 punti] Un corso di laurea desidera gestire automaticamente l'assegnazione delle tesi ai propri studenti, utilizzando le classi C++ di seguito specificate:

```
class Tesi {
    private: string titolo;
    public: string get_titolo(); void set_titolo(string s); };
class Studente {
    private: string nome; Tesi* tesi;
    public: string get_nome(); string set_nome();
           void set_nome(string s); void set_tesi(Tesi* t); };
class Assegnazioni {
    private:
        Tesi* tesi; int n_tesi; int max_tesi;
        Studente* laureandi; int n_laur; int max_laur;
    public:
        Assegnazioni(int max_t, int max_l); ~Assegnazioni();
        void aggiungi_tesi(string titolo_tesi);
        void aggiungi_studente(string nome_studente, string titolo_tesi);
        void laureato(string nome_studente);
        void libera_tesi(string titolo_tesi);
        int disponibili();
}
```

I campi `tesi` e `laureandi` puntano ad arrays contenenti rispettivamente le tesi proposte dai docenti e gli studenti che hanno avuto una assegnazione di tesi. Entrambi gli array sono gestiti come liste. Ciascun studente avrà il proprio campo puntatore collegato ad un oggetto nell'array `tesi`, rappresentante la tesi a lui assegnata. Il costruttore inizializza un oggetto `Assegnazioni` con entrambe le liste vuote e con capienza massima specificata dagli argomenti. I metodi `aggiungi_tesi` e `aggiungi_studente` inseriscono una nuova tesi ed un nuovo studente, rispettivamente, nel sistema. Se la tesi associata allo studente non è già presente nel sistema, questa dovrà essere creata. Il metodo `laureato` elimina dal sistema uno studente e la propria tesi. Il metodo `libera_tesi` elimina lo studente associato alla tesi specificata come argomento, rendendo di nuovo disponibile la tesi stessa. Il metodo `disponibili` restituisce il numero di tesi presenti nel sistema non ancora assegnate ad un laureando. Sviluppare tutti e soli i metodi della classe `Assegnazioni`.