

Compito di Fondamenti di Informatica

(programma C++)

20 dicembre 2005

1. [4 punti] Eseguire le seguenti conversioni, indicando i passaggi intermedi.
 - (a) Convertire il numero frazionario decimale 129.35 in base due.
 - (b) Convertire in notazione modulo e segno su 10 bit i seguenti numeri decimali: -512 , -337 .
 - (c) Convertire in notazione complemento a due su 7 bit i seguenti numeri decimali: -74 , -47 .
 - (d) Convertire in base dieci i seguenti numeri espressi in notazione complemento a due su 7 bit: 1101001, 0010101.
2. [3 punti] Descrivere l'idea alla base dell'algoritmo ricorsivo di ordinamento *merge-sort*. Considerando il vettore in ingresso [7, 25, 13, 3, 9, 1, 12, 3], riportare tutti i vettori intermedi prodotti dalle varie chiamate ricorsive della funzione *merge-sort*.
3. [3 punti] Nel linguaggio C++ il simbolo & rappresenta tre operatori distinti. Discutere tutti e tre tali operatori, illustrando il loro utilizzo e fornendo alcuni esempi di codice.
4. [6 punti] Specificare una funzione C++ avente il prototipo `short singleton(const int* array, int dim)`. Tale funzione restituisce il valore 1 se ciascun elemento dell'array è presente con una sola occorrenza. In caso contrario la funzione restituisce il valore 0. Ad esempio, per il vettore [12, -4, 3, 1, -4, 7] la funzione restituisce 0, perchè l'elemento -4 appare con più di una occorrenza.
5. [16 punti] Un'agenzia per il turismo desidera gestire automaticamente la prenotazione dei posti per un concerto utilizzando le classi C++ di seguito specificate:

```
class Cliente {
    private: string nome; string tel;
    public: string get_nome(); string get_tel();
           void set_nome(string s); void set_tel(string t); };
class Spettacolo {
    private:
        Cliente* prenotazioni; int n_prenot; int max_prenot;
        Cliente* attesa; int n_attesa; int max_attesa;
    public:
        Spettacolo(Spettacolo &s); // costruttore copia
        ~Spettacolo(); int trova(string nome, string tel);
        void prenota(string nome, string tel);
        void disdici(string nome, string tel);
        boolean incompleto();
}
```

L'array `prenotazioni` di dimensione `max_prenot` contiene i clienti che hanno il posto. L'array `attesa` di dimensione `max_attesa` contiene i clienti in lista d'attesa; tale lista deve essere gestita con la politica primo arrivato, primo servito. Le variabili `n_prenot` e `n_attesa` indicano il numero di clienti nei rispettivi array. Il metodo `trova` restituisce 0 se il cliente specificato ha il posto, 1 se il cliente è in attesa e -1 altrimenti. Il metodo `prenota` inserisce il cliente specificato nell'oggetto (eventualmente in attesa). Il metodo `disdici` rimuove il cliente specificato dall'oggetto. Nel caso venga liberato un posto in `prenotazioni`, viene trasferito il primo cliente da `attesa`. Infine, il metodo `incompleto` restituisce `true` se esiste almeno un cliente che abbia almeno un posto ed almeno una prenotazione in attesa.