

Compito di Fondamenti di Informatica
24 settembre 2005

1. [4 punti] Eseguire le seguenti conversioni, indicando tutti i passaggi intermedi.
 - (a) Convertire il numero decimale 107.275 espresso in notazione posizionale in base dieci, nella equivalente rappresentazione in notazione posizionale in base due.
 - (b) Convertire in notazione complemento a due su 7 bit i seguenti numeri decimali (ove possibile): -74, -61, 0.
 - (c) Convertire in notazione modulo e segno su 10 bit i seguenti numeri decimali: -512, -327.
2. [4 punti] Illustrare l'idea alla base dell'algoritmo di ordinamento per unione (*merge sort*), fornendo alcuni esempi del suo funzionamento e discutendo le prestazioni.
3. [3 punti] Si consideri il seguente frammento di programma C++:

```
int k, *p, a[9];
*a = k = 9;
p = &a[7] - 7;
*p += 3;
cout << &p;
```

Commentare le precedenti istruzioni (nell'ordine indicato) specificando il loro significato e segnalando eventuali errori.

4. [9 punti] Implementare in C++ una funzione predicato **ricorsiva**, avente il prototipo

```
bool check_sort(const string s);
```

La funzione restituisce il valore `true` se e solo se gli elementi di `s` sono alfabeticamente ordinati. Ad esempio, la chiamata `check_sort("abfhmqz")` restituisce 1, mentre la chiamata `check_sort("abfmhqz")` restituisce 0.

5. [15 punti] Una compagnia di assicurazioni automobilistiche desidera gestire uno schedario elettronico utilizzando le classi C++ di seguito specificate:

```
class Cliente {
public: string nome; };
class Auto {
public: int targa; Cliente* proprietario; };
class Schedario {
private:
    Cliente clienti[MAX_C];
    Auto auto[MAX_A];
    int n_clienti; int n_auto;
public:
    bool ins_cliente(string nome_cliente); bool rim_cliente(string nome_cliente);
    bool ins_auto(int targa, string nome_cliente); Cliente max_auto(); };
```

Gli array `clienti` e `auto` devono essere gestiti come liste. Ciascun elemento dell'array `auto` contiene un oggetto di classe `Auto` che punta, mediante il campo `proprietario`, ad un oggetto di classe `Cliente` nell'array `clienti`. Le funzioni `ins_cliente` e `ins_auto` inseriscono nuovi oggetti nello schedario; si assuma che tali oggetti non siano già presenti. La funzione `rim_cliente` rimuove un cliente, cancellando inoltre tutte le auto di cui il cliente è proprietario. La funzione `max_auto` restituisce il cliente proprietario del maggior numero di auto, risolvendo in modo arbitrario casi di parità. Suggerimento: per la funzione `max_auto` usare un array ausiliario di interi, parallelo all'array `clienti`, per effettuare i conteggi.