

Compito di Fondamenti di Informatica
09 aprile 2005

1. [4 punti] Rispondere alle seguenti domande. Nel caso di conversioni, indicare i passaggi intermedi.
 - (a) Convertire il decimale 0.185, espresso in notazione posizionale in base dieci, nella equivalente rappresentazione in notazione posizionale in base due.
 - (b) Convertire in notazione complemento a due su 9 bit i seguenti numeri: -197 , 511 .
 - (c) Convertire in base dieci i seguenti numeri espressi in notazione complemento a due su 8 bit: 11101101 , 00010111 .
 - (d) Eseguire le seguenti operazioni in complemento a due su 7 bit, indicando in quali casi si verifichi la condizione di overflow: $1010111 + 0011011$, $0110111 - 1110111$, $1011100 - 1110111$.
2. [4 punti] Illustrare tramite alcuni esempi le operazioni base in C++ di apertura/chiusura e di lettura/scrittura riferite ad un file su disco. Qual'è lo scopo della funzione `unset()`? Discutere un esempio di impiego di tale funzione, riportando un frammento di codice.
3. [5 punti] Specificare in pseudo-codice un algoritmo **ricorsivo** per calcolare la somma di tutti gli elementi di un array assegnato in ingresso. La somma alternata degli elementi di un array è definita come la somma del primo elemento con l'inverso del secondo elemento, con il terzo elemento, con l'inverso del quarto elemento, ecc. Specificare in pseudo-codice un algoritmo **ricorsivo** per calcolare la somma alternata degli elementi di un array assegnato in ingresso. Per entrambi gli algoritmi, specificare quali argomenti in ingresso/uscita debbano essere definiti nel caso di implementazione in C++.
4. [8 punti] Implementare in C++ una funzione predicato, avente il prototipo `bool permut(const vector<int> v1, const vector<int> v2);`. La funzione restituisce il valore `true` se `v2` è una permutazione degli elementi di `v1`, cioè se `v2` contiene tutti e soli gli elementi di `v1`, con possibile modifica del loro ordine.
5. [13 punti] Un'agenzia assicurativa desidera gestire il proprio schedario clienti utilizzando le classi C++ di seguito specificate:

```
class Scheda {
private: string nome_cliente; string scadenza; int targa_auto;
public: string get_nome(); string get_scadenza(); int get_targa();
        void set_nome(string s); void set_scadenza(string s); void set_targa(int t); };
class Schedario {
private:
        Scheda* contenuto;
        int num_schede; int max_schede
public:
        Schedario(int dim); ~Schedario();
        bool apri(string nome, string scadenza, int targa);
        void scade(string scadenza);
        int stat(string anno);  };
```

Il contenuto dello schedario è rappresentato da un array puntato da `contenuto`, avente dimensione `max_schede`. Tale array deve essere allocato nello *heap* e deve essere gestito come una lista. Le date sono tutte espresse nel formato "09-04-2005". Il costruttore crea uno schedario vuoto avente dimensione `dim`. La funzione `nuovo` inserisce una nuova scheda, facendo attenzione a non creare duplicati, e restituisce un valore booleano per indicare il successo o il fallimento dell'operazione. La funzione `scade` rimuove tutte le schede con scadenza espressa da `scadenza`. La funzione `stat` conteggia il numero di schede con scadenza in un dato anno, rappresentato da una stringa `anno` avente il formato "2005". Sviluppare tutte le funzioni associate alla classe `Schedario`.