

Compito di Fondamenti di Informatica
18 dicembre 2004

1. [4 punti] Rispondere alle seguenti domande. Nel caso di conversioni, indicare i passaggi intermedi.
 - (a) Convertire il decimale 0.035, 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: -217 , -277 .
 - (c) Convertire in base dieci i seguenti numeri espressi in notazione complemento a due su 7 bit: 1010101, 0001111.
 - (d) Eseguire le seguenti operazioni in complemento a due su 8 bit, indicando in quali casi si verifichi la condizione di overflow: $00101101 + 10111100$, $10010101 + 10000011$, $01011100 - 11101110$.
2. [4 punti] Si considerino i seguenti prototipi: `int f1(const int n); int f2(int& n); int f3(int* n); int f4(const int* n); int f5(const int& n);`. Specificare per ciascuna funzione la tipologia di chiamata utilizzata (*call by value* oppure *call by reference*) e descrivere l'effetto ottenuto.
3. [5 punti] Due vettori di interi si dicono equivalenti se contengono gli stessi elementi, senza considerare le ripetizioni. Ad es., [6, 7, 13, 6] e [7, 13, 6, 7] sono equivalenti.
 - (a) Descrivere discorsivamente (o in pseudo-codice) un algoritmo che, dati in ingresso due vettori di interi entrambi di dimensione n , restituisca il valore Booleano `true` se i vettori sono equivalenti, `false` altrimenti. L'algoritmo dovrà verificare, per ciascun elemento di ciascun vettore, se questo sia presente anche nell'altro vettore.
 - (b) Usando la notazione $O()$, specificare quanti accessi complessivi ai vettori in ingresso siano richiesti dall'algoritmo al punto (a).
 - (c) Descrivere in pseudo-codice un algoritmo che risolva il problema al punto (a) utilizzando solamente $O(n \log(n))$ accessi ai vettori.
4. [8 punti] Implementare in C++ una funzione **ricorsiva**, avente il prototipo `int stat(const char* str, char ch);`, che restituisca il numero di occorrenze del carattere `ch` nella stringa `str`. Ad esempio, la chiamata `stat("abracadabra", 'b')` restituisce il valore 2, poichè il carattere 'b' appare due volte nella stringa in ingresso. **Non utilizzare** variabili globali o variabili statiche.
5. [14 punti] Una segreteria universitaria desidera gestire un archivio dei propri neolaureati in attesa di assunzione, utilizzando le seguenti classi:

```
class Laureato {
private: string nome; int voto_laurea; string data_laurea;
public: string get_nome(); int get_voto(); string get_data();
        void set_nome(string s); void set_voto(int v); void set_data(string s);};
class Archivio {
private: int n_laureati; int max_laureati; Laureato* laureati;
public:
        Archivio(const Archivio& a); // costruttore di copia
        ~Archivio();
        bool inserisci(string nome, int voto, string data);
        void elimina(string nome);
        int stat(int min, int max);};
```

Per ciascun neolaureato, il voto di laurea è espresso in 110-imi e la data è espressa nel formato "08-12-2004". I neolaureati vengono registrati in un vettore puntato dal campo `laureati` e di capacità massima `max_laureati`. Il vettore deve essere allocato nello *heap* e deve essere gestito come una lista. La funzione `inserisci` registra un neolaureato nella lista solamente se vi è ancora posto e se questi non sia già presente. Vengono restituiti i valori `true` o `false` a seconda che la registrazione abbia successo o meno, rispettivamente. La funzione `elimina` rimuove dalla lista un neolaureato. La funzione `stat` conteggia il numero di laureati nell'archivio aventi voto di laurea compreso tra i due valori assegnati in ingresso. Implementare tutte le funzioni associate alla classe.