

Compito di Fondamenti di Informatica

15 dicembre 2005

1. [4 punti] Rispondere alle seguenti domande, indicando i passaggi intermedi.
 - (a) Convertire i decimali 202.125 e 127.025, entrambi espressi in notazione posizionale in base dieci, nelle equivalenti rappresentazioni in notazione posizionale in base due.
 - (b) Convertire in notazione virgola mobile il seguente numero espresso in singola precisione secondo lo standard IEEE: 01000100 00101010 11000000 00000000.

2. [4 punti] Rispondere alle seguenti domande.
 - (a) Spiegare la differenza tra le seguenti due istruzioni: `char a[] = "Hello"; char* b = "Hello";`.
 - (b) Commentare le seguenti istruzioni:

```
double a[10]; int* p;
cout << a + 1;
p = a + 5; cout << a[p];
p++; cout << *p << " " << &p;
```

3. [5 punti] Specificare in pseudo-codice un algoritmo **ricorsivo** che prenda in ingresso due array di interi, con le rispettive dimensioni, e restituisca il valore booleano `true` se i due vettori sono uguali, ed il valore `false` in caso contrario.

Se i due vettori in ingresso hanno dimensione n , quali sono il minimo ed il massimo numero di chiamate ricorsive che possono essere effettuate su tali dati per la funzione specificata?

4. [8 punti] Implementare in C++ una funzione avente il prototipo `string strdel(const string str1, const string str2);`. Tale funzione cerca all'interno di `str1` la occorrenza più a sinistra di `str2` e la cancella, restituendo la stringa risultante. Se non è presente alcuna occorrenza di `str2`, viene restituita `str1`. Ad esempio, `strdel("abracadabra", "bra")` restituisce `"acadabra"` e `strdel("abracadabra", "bar")` restituisce `"abracadabra"`.

5. [13 punti] Si vogliono gestire le prenotazioni su voli aerei di linea utilizzando le classi C++ di seguito specificate:

```
class Cliente {
private:
    string cod_cliente;
public:
    string get_cod(); void set_cod(string c);    };

const int MAX_P = 200; const int MAX_A = 30;
class Volo {
private:
    string cod_volo;           // codice volo
    int posti_assegnati;       // numero clienti con posto assegnato
    Cliente posti[MAX_P];     // array posti assegnati
    int clienti_attesa;        // numero clienti attesa
    Cliente attesa[MAX_A];    // array lista attesa
public:
    Volo(); Volo(const Volo &v);
    void prenota(string cod_c);
    void cancella(string cod_c);    };
```

La funzione `prenota` inserisce nel volo il cliente avente il codice specificato (array `posti`), se vi è disponibilità, altrimenti inserisce il cliente in una lista d'attesa (array `attesa`). La lista d'attesa è gestita con la disciplina primo arrivato, primo servito. La funzione `cancella` elimina dal volo il cliente avente il codice specificato, rimpiazzando eventualmente con il primo cliente nella lista d'attesa e aggiornando quest'ultima. Non devono mai essere create duplicazioni di clienti nel sistema. Sviluppare tutte le funzioni associate alla classe.