

## Prova scritta del 14/9/2010

*Strutturare adeguatamente i programmi ed evidenziarne la strutturazione mediante indentazione. Inserire anche adeguati commenti*

1) (a) Realizzare una funzione booleana di nome `Inizio` che, presi come suoi parametri un array di caratteri `T` ed una posizione `p` all'interno di `T`, determina se il carattere `T[p]` costituisce o meno l'inizio di una frase, secondo il seguente criterio: un carattere costituisce l'inizio di una frase se e' il primo carattere del testo o se e' preceduto da uno *spazio* o da un *"a capo"* e questi sono a loro volta preceduti da un carattere delimitatore di frase (e cioè uno tra i caratteri `'.'`, `'!'`, `'?'`).

(b) Scrivere un programma principale che legge da `std input` una sequenza di caratteri (max. 1000), terminata da *"a capo"*, e, utilizzando obbligatoriamente la funzione `Inizio`, determina e stampa tutte le posizioni di inizio frase presenti nella stringa. Ad esempio, data la stringa `"CIAO. COME STAI? BENE."`, vengono stampate le posizioni `1`, `7`, e `18`.

(c) Descrivere la funzione `Inizio` anche tramite un diagramma di flusso.

2) Sia `Elem` il tipo di una struttura (`struct`) costituita da due campi, `c1` di tipo reale, e `c2` di tipo stringa (array di caratteri), di lunghezza massima 32. Scrivere un programma principale che legge da due file distinti, di nome `"alfa.txt"` e `"beta.txt"`, due sequenze di elementi di tipo `Elem` e per ogni coppia di elementi corrispondenti (= stessa posizione nei due file) scrive su un terzo file, di nome `"gamma.txt"`, la somma dei due campi `c1` se i campi `c2` sono identici, oppure il massimo tra i due campi `c1` se i campi `c2` sono diversi. La lettura termina appena si raggiunge l'*end-of-file* di uno dei due file di input. SUGG.: non memorizzare gli elementi letti dai file in un array, ma eseguire direttamente su essi le operazioni richieste.

3) Scrivere un programma principale che legga da standard input una sequenza di 25 numeri interi e li memorizzi uno alla volta in una matrice `M` di `5x5` elementi. Il programma quindi costruisce una nuova matrice `B` costituita da soli 0 e 1, di dimensione `5x5`, nel modo seguente: per ogni elemento `Mij` di `M`, se la somma dei valori degli elementi "adiacenti" è maggiore di 10 allora il corrispondente elemento di `B`, `Bij`, conterrà 1; altrimenti `Bij` conterrà 0. Per elementi "adiacenti" ad un elemento `e` si intendono quelli che hanno indice di riga o indice di colonna (ma non entrambi) minore o maggiore di un'unità rispetto a quelli di `e` (N.B. un elemento può avere da un minimo di 2 ad un massimo di 4 elementi adiacenti). Il programma termina stampando su standard output la matrice `B` (una riga di stampa per ogni riga della matrice).

4) [SOLO CdL in FISICA] Date le seguenti istruzioni C++

```
struct elem {int info1; int info2;};
elem* p[10];
p[0] = new elem;
p[0]->info1 = 5;
cout << (*p[0]).info1 + 1;
```

qual è il risultato della loro esecuzione? Giustificare dettagliatamente la risposta. Illustrare anche mediante un disegno le strutture dati create.