

FONDAMENTI DI INFORMATICA L-A
Ing. Gestionale/dei Processi Gestionali – Prof. Penzo
Prova Scritta del 18/02/2011

IMPORTANTE: *condizione necessaria (ma non sufficiente)* al superamento della prova è il corretto svolgimento del punto 1) dell'esercizio 1 e di tutto l'esercizio 2.

ESERCIZIO 1

Si scriva un programma C per la gestione di un servizio quotidiano di messaggistica su telefono cellulare. Il programma gestisce **(2 punti)**:

- **Notizia**, caratterizzati da: tipo di notizia (a soli valori: sport, politica, cronaca), testo della notizia (massimo 5 righe di testo, ognuna di massimo 20 caratteri).
- **Clienti**, caratterizzati da: numero di cellulare, tipo di notizia a cui il cliente è interessato, numero massimo di notizie che il cliente vuole ricevere giornalmente, credito telefonico (espresso in euro).

Ogni notizia ha un costo di 20 centesimi. Le notizie sono memorizzate in un file binario NOTIZIE.DAT, i clienti in un vettore CLIENTI (max 10000 elementi), dove si assuma che i numeri di cellulare siano tutti distinti. Si assuma che il vettore e il file siano già inizializzati con un numero imprecisato di elementi. Si scriva un programma C che definisca e utilizzi le seguenti funzioni/procedure, all'interno delle quali *non è consentito l'uso di variabili globali né l'immissione di valori da standard input all'interno di procedure e funzioni* (**N.B.: Svolgere tutti i punti nel modo più efficiente**):

- 1) procedura *inserisciCliente* che dato un cliente inserisca, se possibile, il cliente nel vettore restituendo il numero eventualmente aggiornato di elementi in esso presenti. **(2 punti)**
 - 2) funzione *notiziePerTipo* che, dato un tipo di notizie, generi un vettore contenente le notizie del tipo dato. La funzione deve restituire il numero di notizie riportate nel vettore. Il vettore da generare e il numero massimo di elementi che esso può contenere sono parametri della funzione. Nel caso si esaurisca lo spazio disponibile, la funzione deve stampare un messaggio. Il nome del file delle notizie è un parametro della funzione. **(4 punti)**
 - 3) procedura *notiziePerCliente* che, dato un cliente, determini le notizie di interesse per il cliente richiamando la funzione del punto 2) per un vettore di massimo 20 notizie. La procedura deve poi generare un file di testo contenente il testo delle prime N notizie di interesse, con N tale che venga rispettato il vincolo sul numero massimo di notizie richieste dal cliente e compatibilmente con il credito disponibile. Per generare il file con le righe di testo, utilizzare la funzione `fputs(<stringa>,<puntatore-a-file>)`. La procedura deve restituire il costo totale per la ricezione delle notizie. Il nome del file da generare deve essere passato come parametro. **(5 punti)**
 - 4) funzione *invioNotizieACliente* che, dato un numero di cellulare, richiami la procedura del punto 3) con il file NOTIZIE.DAT per ottenere il file delle notizie da inviare al cliente, stampi il testo delle notizie in esso contenute (leggendo le righe di testo dal file con la funzione `fgets(<stringa>,<num-stringhe-da-leggere>,<puntatore-a-file>)`) e aggiorni il credito del cliente corrispondente, decrementandolo del costo per la ricezione delle notizie (valore restituito dalla procedura del punto 3) richiamata prima). La funzione deve restituire un valore booleano che attesti l'aggiornamento del credito. **(5 punti)**
- Il programma principale deve, infine, svolgere il seguente punto:
- a) richiedere un cellulare e il nome del file delle notizie da generare per il cliente con quel cellulare e richiamare la funzione del punto 4) stampando l'esito dell'aggiornamento del credito. **(1 punto)**
- 5) Si discuta e si mostri inoltre la sequenza dei record di attivazione sullo stack per l'esecuzione della funzione del punto 3), assumendo che venga richiamata dal programma principale. **(4 punti)**

ESERCIZIO 2

Illustrare il più esaurientemente possibile gli argomenti:

- 1) Il sistema operativo: struttura e funzionalità **(5 punti)**
- 2) Il passaggio dei parametri nelle funzioni/procedure in C **(5 punti)**