



Informatica A – a.a. 2012/2013 – 27/02/2013

Cognome	_____	Matricola	_	_____
Nome	_____	Firma dello studente	_____	

Istruzioni

- Non separate questi fogli. Scrivete la soluzione **solo sui fogli distribuiti**, utilizzando il retro delle pagine in caso di necessità. **Cancellate le parti di brutta** (o ripudiate) con un tratto di **penna**.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- **È possibile scrivere a matita** (e non occorre ricalcare al momento della consegna!).
- È **vietato** utilizzare **calcolatrici, telefoni o pc**. Chi tenti di farlo vedrà **annullata** la sua prova.
- **Non è ammessa la consultazione di libri e appunti**.
- Qualsiasi **tentativo** di comunicare con altri studenti comporta **l'espulsione** dall'aula.
- È possibile **ritirarsi senza penalità**.
- Non è possibile lasciare l'aula conservando il tema della prova in corso.
- Tempo a disposizione: **2h**

Valore indicativo degli esercizi, voti parziali e voto finale:

Esercizio 1 (5 punti)

Esercizio 2 (5 punti)

Esercizio 3 (6 punti)

Esercizio 4 (5 punti)

Esercizio 5 (7 punti)

Totale: (completo 28 punti)

Esercizio 1 - Algebra di Boole, Aritmetica Binaria, Codifica delle Informazioni

(a) Si costruisca la tabella di verità della seguente espressione booleana in tre variabili, badando alla precedenza tra gli operatori logici. Eventualmente si aggiungano le parentesi (1 punto).

(not A or B and not C) or (A or C) and not B

(b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri $A = 23$ e $B = -52$, li si converta, se ne calcolino la somma ($A+B$) e la differenza ($A-B$) in complemento a due e si indichi se si genera riporto sulla colonna dei bit più significativi e se si verifica overflow (2 punti).

- (c) Si converta il numero 29.875 in virgola fissa e in virgola mobile ($r = m \times 2^n$, con m e n codificati in binario, sapendo che $1/2 = 0.5$, $1/4 = 0.25$, $1/8 = 0.125$, $1/16 = 0.0625$, $1/32 = 0.03125$, $1/64 = 0.015625$, e $1/128 = 0.0078125$) (1 punto)

Esercizio 2 - Domanda di teoria

Spiegare che cosa si intende per iterazione e per ricorsione. Quando è necessario, o conviene, usarle?

Esercizio 3 - Matlab

Data la matrice $X = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9]$ illustrare cosa restituiscono i seguenti comandi matlab.

1. $X(1, :)$
2. $X([3 \ 1 \ 2], 2) = [1 \ 2 \ 3]$
3. $X(2:end, :)$
4. $X-X'$
5. $X(\text{rem}(X,2)==0) = X(\text{rem}(X,2)==0)+1$

Dato il vettore $X = [15 \ 18 \ 6 \ 21 \ 30 \ 32 \ 25 \ 14 \ 23 \ 17 \ 24 \ 18]$ contenente i voti di un esame, scrivere il codice matlab che:

1. conta il numero di studenti insufficienti ed il numero di 30 e lode.
2. calcola la media dei voti sufficienti
3. somma 3 punti ai voti inferiori al 18

Esercizio 4 - Programmazione C (Ricorsione)

Si supponga di avere a disposizione il tipo `File`, usato per rappresentare sia un file del filesystem che una cartella (in inglese `directory`). Si supponga inoltre di avere le seguenti due funzioni:

- **`int isDirectory(File f)`** - restituisce 1 se il file **`f`** è una `directory`, 0 altrimenti.
- **`int listFiles(File f, File *files)`** - modifica l'array **`files`** (passato come parametro), inserendovi i file che sono "figli" del FILE **`f`** nella gerarchia del filesystem. Restituisce, inoltre, come parametro di ritorno il numero di file che ha inserito in **`files`**. Nel caso in cui **`f`** è una cartella vuota o un file "semplice", la funzione restituisce 0, e in entrambi i casi non viene modificato l'array **`files`**.

Si implementi una funzione RICORSIVA **`int countFiles(File f)`** che conta il numero di file presenti all'interno di un dato **`f`**. La funzione restituisce 1 sia nel caso in cui **`f`** è un file "semplice", sia nel caso in cui **`f`** è una `directory` vuota.

Esercizio 5 - Programmazione in C (Liste Dinamiche)

Si ipotizzi di dovere implementare un programma gestionale per un torneo di calcio. Il programma deve gestire una lista chiamata **classifica**.

La lista contiene squadre di calcio, ordinate dalla prima all'ultima in classifica. Per ciascuna squadra viene memorizzato il nome della squadra, i punti totalizzati, e l'elenco dei giocatori. Per ciascun giocatore, inoltre, viene memorizzato il nome, il cognome, il numero di maglia, e il numero di gol realizzati nel torneo.

Si definiscano le strutture dati necessarie allo sviluppo di questo programma gestionale.

Inoltre, definire e implementare:

1. una funzione per aggiornare la lista **classifica**:

ptrClassifica aggiornaClassifica(ptrClassifica testa, risultatoPartita ris);

NB. **risultatoPartita** è da considerarsi come un tipo di dato che contiene il nome delle due squadre che hanno giocato, il numero di gol realizzati da ciascuna squadra, e i giocatori che hanno segnato.

La funzione deve aggiornare la lista **classifica** nei seguenti modi:

1. deve assegnare tre punti alla squadra vincente, oppure 1 punto a entrambe le squadre se hanno pareggiato
2. deve riordinare la lista se l'aggiunta di punti ha fatto cambiare la posizione in classifica a una delle due squadre
3. deve aggiornare il numero di gol realizzati dai giocatori delle squadre che si sono incontrate