Politecnico di Milano

Dipartimento di Elettronica e Informazione

Informatica A – a.a. 2013/2014 – 01/07/2014

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).

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

(b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri A = 17 e B = −13, 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 30.875 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 cosa sono i puntatori nel linguaggio di programmazione C e a cosa servono. Si faccia uso di esempi.

Esercizio 3 - Matlab

Scrivere il codice Matlab che restituisce i valori richiesti. Attenersi al numero di righe di codice indicato. L'eventuale utilizzo di più righe comporterà una minore valutazione dell'esercizio.

Data la matrice:

```
A =

    32    76    36   -97    75
    16    56    12   -32   -12
    78    -6   -23    55    74
    14   -21    43    83    31
```

1. Creare la matrice A (max 1 riga) (0.5 punti)

2. Eliminare la 3 colonna (max 1 riga) (0.5 punto)

3. Inserire dopo la seconda colonna una colonna contenenti valori casuali da -11 a 37 (max 2 righe) (1 punto)

3. Chiedere un numero all'utente e stampare solamente i numeri di cui il numero inserito è divisore (max 2 righe) (1 punto):

4. Creare una matrice B che contenga i valori di A meno la media dei soli valori negativi (max 1 riga) (1 punto)

5. Scrivere una funzione Matlab che calcoli la somma dei valori interi da 1 fino a n, con n inserito dall'utente. Stampare a video **solamente il valore della somma.** (max 8 righe) (2 punti)

Esercizio 4 - Programmazione C

Scrivere un programma che chiede all'utente di inserire una sequenza (di lunghezza a priori illimitata) di interi positivi. L'inserimento deve continuare fino all'inserimento del numero -1.

Il programma deve stampare a video il numero di terne pitagoriche irriducibili, formate da numeri adiacenti, che sono presenti all'interno della sequenza.

Una terna pitagorica è irriducibile se, oltre a rispettare la classica proprietà pitagorica, i suoi elementi sono a due a due primi tra loro. La terna (3, 4, 5), ad esempio, è irriducibile. La terna (6, 8, 10) è pitagorica ma non irriducibile. Infatti è creata moltiplicando per 2 gli elementi della prima terna.

NB. Non è consentito salvare la sequenza all'interno di un array.

Esercizio 5 - Programmazione in C (Liste Dinamiche)

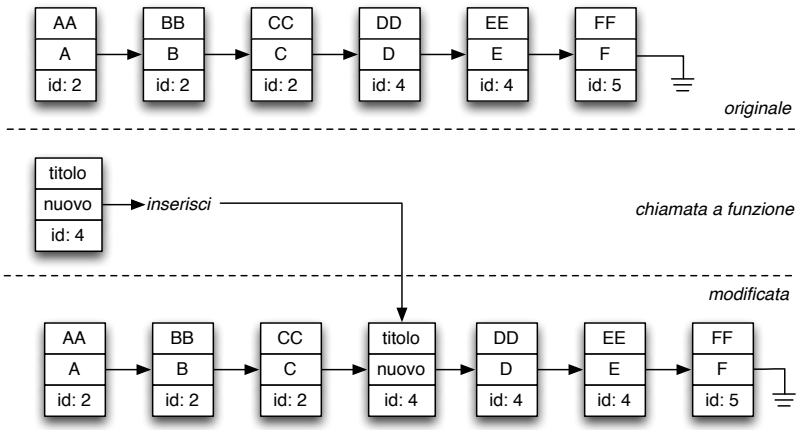
Implementare una TODO list multi-utente.

Una TODO list è una lista di attività che devono essere svolte. Una TODO list multi-utente è una variante in cui a ogni attività è assegnata una persona che la deve eseguire.

Implementare la TODO list mediante una lista di elementi concatenati. Ogni elemento deve contenere un titolo, un testo e un identificativo numerico della persona incaricata. Gli elementi della lista devono essere ordinati prima per identificativo dell'utente (in ordine crescente) e, successivamente (ovvero per elementi che hanno lo stesso identificativo utente), dall'elemento inserito più recentemente a quello inserito da più tempo.

1. Definire le strutture dati necessarie.

2. Implementare una funzione **aggiungiAttività** che prende in ingresso una nuova attività e lo inserisce al suo posto nella TODO list. Si veda l'esempio sottostante.



3. Implementare una funzione **rimuoviAttività** che prende in ingresso un identificativo utente e rimuove l'elemento inserito da più tempo che appartiene a quell'identificativo utente. Restituisce inoltre un puntatore all'elemento eliminato. Si veda l'esempio sottostante.