



Politecnico di Milano

Dipartimento di Elettronica e Informazione

Informatica A – a.a. 2015/2016 – 16/02/2016

Cognome _____ Matricola _____
Nome _____ Firma _____

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** con un tratto di **penna**.
- Ogni parte non cancellata a penna sarà considerata parte integrante della soluzione.
- **NON è possibile scrivere a matita.**
- 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: **2h30m**

Valore indicativo degli esercizi, voti parziali e voto finale:

Esercizio 1	3 punti	_____
Esercizio 2	3 punti	_____
Esercizio 3	8 punti	_____
Esercizio 4	6 punti	_____
Esercizio 5	8 punti	_____

Totale (28): _____

Esercizio 1 - Algebra di Boole, Aritmetica Binaria, Codifica delle Informazioni (3 punti)

- (a) Si costruisca la tabella di verità della seguente espressione booleana in tre variabili, badando alla precedenza tra gli operatori logici. Eventualmente si aggiungano parentesi. Non si accetteranno soluzioni senza il procedimento. (1 punto)

$A \text{ and } C \text{ or not } C \text{ and (not } B \text{ or not } A)$

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri $A = -39_{\text{dec}}$ e $B = 26_{\text{dec}}$, 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. Non si accetteranno soluzioni senza il procedimento. (1 punto)

(c) Si converta il numero 96.3125 in virgola fissa e in virgola mobile con codifica IEEE 754, 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 $1/128 = 0.0078125$. Non si accetteranno soluzioni senza il procedimento. (1 punto)

Esercizio 2 - Domanda di teoria (3 punti)

1. Spiegare in cosa consiste l'iterazione. (1 punto)
2. Spiegare in cosa consiste la ricorsione. (1 punto)
3. Spiegare cosa hanno in comune i due approcci e cosa li differenzia. (1 punto)

Esercizio 3 - Programmazione C (8 punti)

Si considerino poligoni nel piano cartesiano identificati dai loro vertici. I vertici del poligono vengono registrati in due array di uguale lunghezza: il primo di nome `xx` per le ascisse, il secondo di nome `yy` per le ordinate. Entrambi hanno dimensione massima 100, mentre le dimensioni effettive sono specificate dall'utente. Si considerino solamente figure chiuse, quindi si assuma che esista sempre un segmento che unisce l'ultimo vertice al primo.

1. Si definisca e si implementi una funzione per l'acquisizione dei vertici, specificando come parametro il numero di vertici da acquisire. (2 punti)
2. Si definisca e si implementi la funzione `calcolaPerimetro` che prende in ingresso i due vettori `xx` e `yy` e restituisce al programma chiamante:
 - il perimetro `p` del poligono (il perimetro si calcola come la somma delle distanze tra vertici consecutivi) (2 punti), e
 - un valore `r = 1` se il poligono non ha vertici nel secondo e quarto quadrante, `r = 0` altrimenti. (2 punti)
3. Si scriva nel main la chiamata alla funzione `calcolaPerimetro` e si stampino i risultati. (1 punto)

NOTA 1: Per piano cartesiano si intende il piano diviso in quattro parti dai due assi, ciascuna di queste parti è detta quadrante. I quadranti sono ordinati convenzionalmente in senso antiorario a partire da quello delimitato dai semiassi positivi. Punti che giacciono sugli assi non sono di nessun quadrante.

NOTA 2: Per effettuare il calcolo della distanza si utilizzi la funzione `sqrt` della libreria `math`.

Esempi:

Si consideri il seguente triangolo:

```
xx = [0 -1 -1]
yy = [1 1 2]
```

`calcolaPerimetro` restituisce come perimetro 3.414 (approssimazione di $2 + \sqrt{2}$) e il valore `r=0`. Infatti il triangolo ha vertici nel secondo quadrante.

Si consideri il seguente quadrato:

```
xx = [0 2 2 0]
yy = [0 0 2 2]
```

`calcolaPerimetro` restituisce come perimetro 8 e il valore `r=1`. Infatti il quadrato non ha vertici nel secondo o quarto quadrante.

4 - Matlab (6 punti)

Scrivere il codice Matlab che restituisca i valori richiesti. Attenersi al numero massimo di righe di codice indicato. Tutti gli esercizi sono risolvibili con il numero di righe indicato; un numero maggiore di righe utilizzato rispetto a quello indicato comporterà una valutazione inferiore. La soluzione deve essere generalizzata e non applicata ad una data matrice. Due comandi consecutivi sono considerate due righe (es.: $A = A * A$; $A = A'$).

1. Creare la matrice A di numeri casuali tra -20 e 30 di dimensione 5 righe e 7 colonne (1 riga) (1 punti):
2. Calcolare il valore medio totale della sole righe dispari(1 riga) (1 punto)
3. Eliminare le colonne la cui media è inferiore alla media totale della matrice(1 riga) (1.5 punto)
4. Moltiplicare ogni cella di A per la radice quadrata della media di A (1 riga) (1 punto)
5. Scrivere una funzione Matlab ricorsiva $[out] = ricorsiva(A,x)$ che restituisca la somma delle righe pari meno la somma delle righe dispari. (max 11 righe) (1,5 punto)

Esercizio 5 - Programmazione C (8 punti)

Si ipotizzi di dovere scrivere un programma per BikeMi. Questo programma deve permettere l'immagazzinamento di informazioni sull'uso delle biciclette in città, e deve permettere a BikeMi di estrarre informazioni utili per il miglioramento del servizio offerto.

Le informazioni sono immagazzinate in una lista. Ciascun elemento della lista rappresenta un singolo utilizzo di una bici e contiene le seguenti informazioni: quando è iniziato l'utilizzo, quando è finito l'utilizzo, un identificativo della stazione di partenza, un identificativo della stazione di arrivo, un identificativo della bici utilizzata, e l'anno di nascita della persona che ha usato la bici.

Nota: I momenti di inizio e di fine dell'utilizzo devono essere salvati come "numero di secondi dal 1° Gennaio del 1970";

1. Definire le strutture dati necessarie a creare questa lista. (1 punto)
2. Si definisca e si implementi una funzione per l'inserimento di nuovi elementi nella lista. Gli elementi nella lista devono essere ordinati per momento di inizio dell'utilizzo. (2 punti)

Si ipotizzi adesso che BikeMi voglia analizzare l'uso che diverse generazioni di utenti fanno del loro servizio. Vengono individuate sei fasce di età: 16-19, 20-29, 30-39, 40-49, 50-59 e 60-69.

Nota: si ipotizzi che il servizio venga offerto solo a persone che hanno tra 16 e 69 anni.

3. Si definisca e si implementi una funzione che prende in ingresso la lista originale e genera un array di sei elementi del tipo "fasciaEtà". Ciascun elemento del tipo fasciaEtà contiene al suo interno l'età minima per appartenere a quella fascia, l'età massima per appartenere a quella fascia, il numero di utilizzi effettuati da persone di quella fascia di età, la durata media degli utilizzi effettuati da persone di quella fascia di età, e un puntatore a una lista che contiene tutti gli utilizzi fatti da persone di quella fascia di età. Questa lista non deve seguire alcun ordinamento particolare. (1 punto per la definizione del tipo fasciaEtà e 3 punti per la funzione)

Nota: Le informazioni necessarie a creare i sei elementi di tipo fasciaEtà devono essere prese dalla lista originale.

Nota 2: La creazione di questi sei elementi non deve causare effetti collaterali sulla lista originale!

