



Politecnico di Milano

Dipartimento di Elettronica e Informazione
Informatica A a.a. 2011/2012 15/11/2010

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** (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 (2 punti)

Esercizio 2 (2 punti)

Esercizio 3 (3 punti)

Esercizio 4 (3 punti)

Esercizio 5 (4 punti)

Totale: (14 punti)

Esercizio 1 - Algebra di Boole, Aritmetica Binaria, Codifica delle Informazioni (2 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 le parentesi. Non si accetteranno soluzioni senza il procedimento.

not (A and notB or C) and (notA or C)

(b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri $A = 62_{\text{dec}}$ e $B = -20_{\text{dec}}$, 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.

(c) Si converta il numero 53.375 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 $1/128 = 0.0078125$). Non si accetteranno soluzioni senza il procedimento.

Esercizio 2 - Architettura del calcolatore (2 punti)

Si descriva brevemente cosa si intende per memorie di tipo RAM e di tipo ROM, e si indichi a cosa servono.

Esercizio 3 - Interpretazione del codice (3 punti)

Completare il seguente codice con il proprio nome (variabile n) e la propria matricola (variabile m). Indicare e spiegare cosa viene stampato a video.

```
#include <stdio.h>
#include <string.h>
#define MAX 10
```

```
int main ()
{
    char n[MAX] = "_____";

    int m = _____;

    int c[MAX];
    int l, i, mm, d;

    printf("n e' %s.\n", n);

    mm = (m%1000)/100;
    if (mm%2==0) {
        d = 1;
    }
    else {
        d=2;
    }
    printf("d is %d.\n", d);

    l = (int)strlen(n);
    printf("l e' %d.\n", l);

    for (i=0; i<l; i++) {
        c[i] = n[i];
    }

    printf("output e' ", l);
```

```
for (i=0; i<l; i++) {
    if (i==0) {
        if (c[0] > c[1]) {
            printf("%c", (char)(c[0]+d));
        }
        else {
            printf("%c", c[0]);
        }
    }
    else if (i==l-1) {
        if (c[l-1] > c[l-2]) {
            printf("%c", (char)(c[l-1]+d));
        }
        else {
            printf("%c", c[l-1]);
        }
    }
    else {
        if (c[i]>c[i-1] && c[i]>c[i+1]) {
            printf("%c", (char)(c[i]+d));
        }
        else {
            printf("%c", c[i]);
        }
    }
}

return 0;
}
```

Esercizio 4 - Stringhe (3 punti)

Nella libreria di funzioni `<string.h>` esiste una funzione chiamata **strcmp** che confronta due stringhe e restituisce un numero minore di zero se la prima stringa viene prima della seconda in ordine alfabetico, uguale a zero se le due stringhe sono uguali, o un numero maggiore di zero se la prima stringa viene dopo la seconda in ordine alfabetico.

Si fornisca una possibile implementazione della funzione strcmp. Si scriva, dunque, un programma che prima chiede all'utente di inserire due stringhe, e poi stampa a video un numero intero usando la stessa strategia della funzione strcmp.

Esercizio 5 - Matrici (4 punti)

```
typedef enum direzione {su, giu, destra, sinistra} Direzione;
typedef struct {
    Direzione d;
    int lunghezza;
} linea;
typedef linea disegno[10];
```

Ipotizzare di avere una matrice di caratteri di dimensione 25x25, inizialmente riempita interamente del carattere `.`. Scrivere un programma che, dato un disegno espresso come un array di al più 10 linee, parte dalla posizione (12,12) e lo disegna. Ciascuna linea ha una direzione e una lunghezza che equivalgono rispettivamente alla direzione di disegno rispetto al punto della matrice in cui il programma si trova in quel momento, e al numero di celle nella matrice che la nuova linea deve occupare. Il disegno di una linea consiste nel riempire un determinato numero di celle della matrice con il carattere `X`. Il programma deve terminare con la stampa a video dell'intera matrice.

Qualora il disegno comportasse un errore, ovvero una linea che esce dai confini della matrice, il programma deve segnalare l'errore all'utente mediante un messaggio sul monitor.

In figura potete vedere un esempio in cui vengono mostrate solo le celle in un intorno della posizione (12,12) e il disegno di un quadrato di dimensione quattro per quattro. Le linee del disegno sono: (su, 4)/(destra,4)/(giu, 4)/(sinistra,4).

					X	X	X	X			
					X			X			
					X			X			
					X	X	X	X			

