



Politecnico di Milano

**Dipartimento di Elettronica e Informazione**  
Informatica A a.a. 2011/2012 10/09/2012

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 ( 4 punti )**

**Esercizio 2 ( 6 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).

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

(b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri  $A = 23$  e  $B = -49$ , 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 56.3125 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 - Interpretazione del codice C

Si consideri il seguente codice:

```
int main()
{

    int i, j, k;
    int temp;
    int iMin;
    int a[N] = {9, 5, 2, 7, 6, 1, 3, 4, 8, 0};

    for (j = 0; j < N-1; j++) {
        iMin = j;
        for ( i = j+1; i < N; i++) {
            if (a[i] < a[iMin]) {
                iMin = i;
            }
        }
        if ( iMin != j ) {
            temp = a[iMin];
            a[iMin] = a[j];
            a[j] = temp;
        }
    }

    for (k=0; k<N; k++) {
        printf("%d ", a[k]);
    }

    return 0;
}
```

Si dica che cosa fa il programma sovrastante.

Spiegare come lo fa e illustrare “passo per passo” come viene modificato il contenuto dell’array a[N].

### Esercizio 3 - DB

Sia data la seguente basi di dati per la gestione di un catalogo a punti (come quelli dei supermercati):

Premio (IdPremio, Descrizione, PuntiNecessari)

Utente (IdUtente, Nome, PuntiDisponibili)

ConsegnaPremio (IdConsegna, IdUtente, IdPremio, Data)

1. Scrivere la query SQL che estrae l'id utente e il nome delle persone che hanno ritirato il premio "Friggitrice" nel mese di Luglio di quest'anno. Riportare per ciascuno anche la data in cui ha ritirato il premio.
2. Scrivere la query SQL che estrae l'id utente e il nome delle persone che hanno ritirato premi per più di 50000 punti nel periodo Giugno-Luglio di quest'anno.

#### **Esercizio 4 - Domanda di Teoria**

Spiegare cosa si intende per "iterazione" e per "ricorsione". Illustrare a cosa servono e discutere le differenze tra i due approcci.

## Esercizio 5 - Programmazione in C

Si consideri il seguente programma:

```
int main(int argc, const char * argv[])
{
    char a[100], b[100];
    int flag = 0;

    printf("Inserire la prima stringa: \n");
    scanf("%s", a);

    printf("Inserire la seconda stringa: \n");
    scanf("%s", b);

    flag = anagramma(a, b);

    if (flag == 1)
        printf("\"%s\" e \"%s\" sono anagrammi.\n", a, b);
    else
        printf("\"%s\" e \"%s\" non sono anagrammi.\n", a, b);

    return 0;
}
```

Implementare la funzione **int anagramma(char a[], char b[])** che prende in ingresso due stringhe e restituisce 1 se sono anagrammi, 0 altrimenti. Due stringhe sono anagrammi se ciascuna può essere ricavata per mezzo della permutazione dei caratteri dell'altra.

**NB. La soluzione deve fare uso di array. Non si accettano soluzioni in cui questi ultimi sono sostituiti da un numero "enorme" di variabili.**