



Informatica A – a.a. 2010/2011 – 08/02/2011

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:

Esame completo: esercizi 1-5

Esercizio 1 (completo 4 punti)

Esercizio 2 (completo 8 punti)

II Compitino: esercizi 3-5

Esercizio 3 (completo 5 punti | | II compitino 5 punti)

Esercizio 4 (completo 4 punti | | II compitino 4 punti)

Esercizio 5 (completo 7 punti | | II compitino 5 punti)

Totale: (completo 28 punti | II compitino 14 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).

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

(b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri $A = 53_{\text{dec}}$ e $B = -14_{\text{dec}}$, 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 (2 punti).

(c) Si converta il numero 9.75 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 - Programmazione in C

Descrivere il funzionamento del seguente codice C, indicando precisamente cosa viene stampato a video:

```
#define N 8

int main(void) {
    int matr[N][N];
    int i,j;
    for (i=1; i<=N/2; i++) {
        for (j=i-1; j < N-(i-1); j++ ) {
            matr[i-1][j] = i;
            matr[N-i][j] = i;
            matr[j][i-1] = i;
            matr[j][N-i] = i;
        }
    }
    for (i=0; i<N;i++) {
        for (j=0; j<N; j++) {
            printf(" %d ", matr[i][j]);
        }
        printf("\n");
    }
}
```

Esercizio 3 - DB

Sia data la seguente basi di dati per un social network:

UTENTE (Id, Nome, Cognome, Sesso)

AMICI (Id_1, Id_2, DataInizioAmicizia)

RELAZIONE(Id_1, Id_2, TipoRelazione, DataInizioRelazione)

Si ipotizzi che per ciascuna relazione di amicizia ci sia una sola riga nella tabella AMICI. Id_1 rappresenta la persona che ha fatto la richiesta di amicizia, mentre Id_2 rappresenta la persona che ha accettato la richiesta di amicizia.

Si ipotizzi inoltre che le relazioni presenti nella tabella RELAZIONE non siano "simmetriche". Nella colonna Id_1 viene registrato l'id della persona che ha stabilito la relazione. Dunque, se in una riga della tabella RELAZIONE Id_1 è in una relazione "complicata" con Id_2, non è detto che Id_2 la pensi allo stesso modo. Potrebbe esserci una riga in cui Id_2 è presente come valore della colonna Id_1 e Id_1 è presente come valore della colonna Id_2, ma la relazione è di tipo diverso. Oppure, potrebbe succedere che Id_2 non abbia stabilito alcun tipo di relazione con Id_1.

1. Scrivere in SQL l'interrogazione che estrae il nome e il cognome di ciascun utente, e il numero di richieste di amicizie da lui/lei effettuate nel 2010.
2. Scrivere in SQL l'interrogazione che estrae il nome e il cognome di quelle persone che hanno stabilito più di 3 relazioni "complicate" nel 2010.

Esercizio 4 - Domanda di Teoria

Spiegare come è possibile rendere permanente su file una lista dinamica di strutture dati definite dal programmatore.

Quali sono i passi che deve compiere il codice che salva la lista su file? e il codice che invece recupera la lista da file?

Esercizio 5 - Ricorsione

Si consideri il seguente codice:

```
typedef struct EL {
    int dato;
    struct EL *left;
    struct EL *right;
} Nodo;
typedef Nodo *ptrNodo;

ptrNodo creaAlbero(int *a, int inizio, int fine) {
    ptrNodo temp;
    int pos;
    int i;

    if ( *** CONDIZIONE CASO BASE **** ) {
        return NULL;
    }

    pos = (fine+inizio)/2;

    temp = (ptrNodo)malloc(sizeof(Nodo));
    temp->dato = *(a+pos);

    temp->left = creaAlbero(*** PARAM PRIMO PASSO INDUTTIVO ***);
    temp->right = creaAlbero(*** PARAM SECONDO PASSO INDUTTIVO ***);

    return temp;
}
```

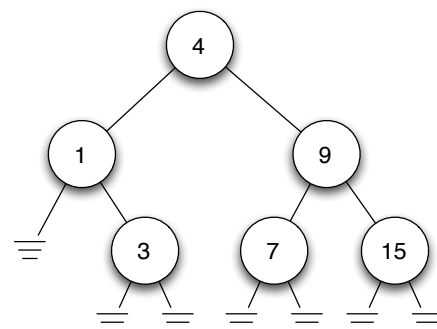
Il codice definisce una funzione ricorsiva creaAlbero che prende in ingresso un array ordinato di numeri e costruisce un albero binario che rispetta la seguente proprietà:

- scelto un nodo qualsiasi, il valore in esso contenuto è maggiore di tutti i valori contenuti nel suo sotto-albero sinistro e minore di tutti i valori contenuti nel suo sotto-albero destro

Ad esempio, a partire dall'array

`int a[N] = {1, 3, 4, 7, 9, 15}`

la funzione produce l'albero in figura.



- 1) Spiegare la logica dietro la funzione creaAlbero
- 2) Indicare come deve essere fatta la prima chiamata alla funzione, ovvero spiegare quali parametri devono essere passati alla prima invocazione
- 3) Completare il codice indicando la condizione che determina il caso base della ricorsione
- 4) Completare il codice indicando i parametri da usare nei due passi induttivi