

**PROVA SCRITTA DEL MODULO DI
ELEMENTI DI INFORMATICA
CORSO DI LAUREA IN INGEGNERIA BIOMEDICA
24 gennaio 2020**

MOTIVARE IN MANIERA CHIARA LE SOLUZIONI PROPOSTE A CIASCUNO DEGLI ESERCIZI SVOLTI

Esercizio 1 (4 punti)

Scrivere la tabella di verità della seguente espressione booleana:

$$F = A \oplus B + A \cdot B$$

Indicare se esiste un'altra espressione booleana coincidente con F e scrivere quale.
Nota: il simbolo \oplus corrisponde all'operatore XOR.

Esercizio 2 (4 punti)

Descrivere in modo chiaro e sintetico il modulo di gestione della memoria in un moderno sistema operativo.

Esercizio 3 (25 punti)

(8 punti) Il Ministero dell'Università e della Ricerca ha deciso di classificare tutti i professori universitari inserendoli in un archivio nel quale, accanto al loro numero di matricola, viene indicato il loro *livello di reputazione*, che può essere *Alto*, *Medio*, *Basso*. Tale valutazione viene fatta ogni 5 anni in funzione di tre criteri: (1) numero di articoli scientifici pubblicati sulle maggiori riviste internazionali, (2) ammontare dei finanziamenti in euro portati nei relativi atenei attraverso progetti, (3) qualità media della didattica svolta in aula, secondo i questionari di valutazione compilati dagli studenti, sintetizzata da un punteggio compreso tra 0 e 10.

L'associazione del livello di reputazione in funzione di detti numeri viene fatta per mezzo della seguente tabella, dove sono riportati i valori minimi ammessi per ciascuna voce. Per entrare nel livello di reputazione associato bisogna che **almeno due** criteri siano soddisfatti secondo la regola dello "strettamente maggiore di":

Livello di reputazione	di	Numero di articoli scientifici	Finanziamenti in euro	Qualità della didattica
Alto		10	100.000/00	8
Medio		5	30.000/00	5
Basso		1	5.000/00	3

I docenti che non rientrano in nessuna delle tre categorie sono a rischio licenziamento, in caso di diminuzione di fondi assegnati al ministero per pagare i relativi stipendi.

Alla fine di ciascun quinquennio di valutazione, ogni ateneo stila un file "docenti.txt" nel quale, in ogni riga, vengono indicati: il numero di matricola del docente, il numero di articoli da lui pubblicati, l'ammontare dei finanziamenti da lui prodotto, la qualità della didattica da lui erogata.

A questo punto vi candidate come consulenti del ministero per scrivere un programma che, ricevendo in ingresso il file di cui sopra, stamperà un nuovo file

"reputazione.txt" in cui verrà riportato in ogni riga il numero di matricola ed il livello di reputazione di ciascun docente, per livello decrescente di reputazione (da "Alto" a "Basso"). In un secondo file, "licenziabili.txt", verrà inserito il numero di matricola di quei docenti ai quali non è stato possibile associare alcun livello di reputazione.

Per implementare il programma, vi servite delle seguenti strutture dati:

```
typedef struct
{
    char matricola[20];
    int nriviste, qdidattica;
    float finanziamenti;
} Docente;
```

```
typedef struct atomo
{
    Docente d;
    struct atomo *successivo;
} ListaDocenti;
```

Le funzioni `CONS`, `HEAD`, `TAIL`, `ISEMPTY` per l'accesso alle variabili di tipo `ListaDocenti` sono già state implementate.

Scrivete inoltre le seguenti funzioni:

1. (4 punti) `leggiDocente(f)`: legge una riga del file aperto `f` strutturato secondo il formato di "docenti.txt" e restituisce una variabile di tipo `Docente` i cui slot sono assegnati in base ai valori letti nella riga.
2. (6 punti) `decidiReputazione(d)`: stabilisce il livello di reputazione di un docente i cui dati sono forniti attraverso la variabile `d` di tipo `Docente`. Il valore restituito è 3 se il livello di reputazione è alto, 2 se medio, 1 se basso, 0 se non è possibile associare alcun livello di reputazione.
3. (7 punti) `stampaFile(nomefile, l, r)`: per ogni riga del file di nome `nomefile`, da aprire in modalità opportuna, stampa il numero di matricola presente negli elementi della lista `l` di tipo `ListaDocenti` e la stringa `r` relativa al livello di reputazione. Se `r` è `NULL`, viene stampato solo il numero di matricola.

Nota: implementare le funzioni `CONS`, `HEAD`, `TAIL`, `ISEMPTY` per l'accesso alle variabili di tipo `ListaDocenti` può fruttare un bonus di massimo 5 punti.

Soluzione

Esercizio n.1

Dalle seguenti tabelle di verità si ha:

A	B	A XOR B	A AND B	F	A OR B
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	0	1	1
1	1	0	1	1	1

E' evidente pertanto che F ha la stessa espressione di A OR B, con la quale coincide.

Esercizio n.2

Si rimanda alle dispense del corso.

Esercizio n.3

```
#include <stdio.h>
#include <stdlib.h>

//Strutture dati fornite nel testo
typedef struct
{
    char matricola[20];
    int nriviste, qdidattica;
    float finanziamenti;
} Docente;

typedef struct atomo
{
    Docente d;
    struct atomo *successivo;
} ListaDocenti;

//Funzioni per l'accesso a ListaDocenti
ListaDocenti* CONS(ListaDocenti *l, Docente d)
{
    ListaDocenti *q;

    q=(ListaDocenti*)malloc(sizeof(ListaDocenti));
    q->d=d;
    q->successivo=l;

    return q;
}
```

```

Docente HEAD(ListaDocenti *l)
{
    return l->d;
}

ListaDocenti* TAIL(ListaDocenti *l)
{
    return l->successivo;
}

int ISEMPY(ListaDocenti *l)
{
    return l==NULL;
}

//Funzioni chieste dal problema
/* leggiDocente(f): legge una riga del file aperto f
strutturato secondo il formato di "docenti.txt" e
restituisce una variabile di tipo Docente i cui slot
sono assegnati in base ai valori letti nella riga.*/

Docente leggiDocente(FILE *f)
{
    Docente d;
    fscanf(f,"%s",&d.matricola[0]);
    fscanf(f,"%d",&d.nriviste);
    fscanf(f,"%f",&d.finanziamenti);
    fscanf(f,"%d",&d.qdidattica);

    return d;
}

```

```

/*decidiReputazione(d): stabilisce il livello di
reputazione di un docente i cui dati sono forniti
attraverso la variabile d di tipo Docente. Il livello
di reputazione è un intero compreso tra 0 e 3 - 0 se
non è possibile associare alcun livello
di reputazione.*/
int livello(Docente d, int lr, float lf, int lq)
{
    int f=d.finanziamenti>lf;
    int r=d.nriviste>lr;
    int q=d.qdidattica>lq;

    return (f&&q)|| (f&&r)|| (q&&r);
}

int decidiReputazione(Docente d)
{
    if(!livello(d,10,100000.,8))
        if(!livello(d,5,30000.,5))
            if(!livello(d,1,5000.,3))
                return 0;
            else
                return 1;
        else
            return 2;

    return 3;
}

/* stampaFile(nomefile,l,r): per ogni riga del file
di nome nomefile, da aprire in modalità opportuna,
stampa il numero di matricola presente negli elementi
della lista l di tipo ListaDocenti e la stringa r
relativa al livello di reputazione.
Se r è NULL, viene stampato solo il numero di
matricola. */
void stampaFile(char* nomefile, ListaDocenti *l, char* r)
{
    FILE *f;
    Docente d;

    f=fopen(nomefile,"a");
    while(!ISEMPTY(l))
    {
        d=HEAD(l);
        fprintf(f,"%s",d.matricola);
        if(r)
            fprintf(f," %s",r);
        fprintf(f,"\n");
        l=TAIL(l);
    }
    fclose(f);
}

```

```

//Funzione main n.1
int main()
{
    Docente d;
    int reputazione;
    ListaDocenti *alto, *medio, *basso, *licenziabili;
    FILE *f;

    alto=NULL;
    medio=NULL;
    basso=NULL;
    licenziabili=NULL;

    f=fopen("docenti.txt","r");
    while(!feof(f))
    {
        d=leggiDocente(f);
        reputazione=decidiReputazione(d);
        switch(reputazione)
        {
            case 3: alto=CONS(alto,d); break;
            case 2: medio=CONS(medio,d); break;
            case 1: basso=CONS(basso,d); break;
            default: licenziabili=CONS(licenziabili,d);
        }
    }
    fclose(f);

    stampaFile("reputazione.txt", alto, "Alto");
    stampaFile("reputazione.txt", medio, "Medio");
    stampaFile("reputazione.txt", basso, "Basso");
    stampaFile("licenziabili.txt", licenziabili, NULL);

    //aggiungere funzioni di deallocazione delle liste
    return 0;
}

```

```

//Funzione main n.2
ListaDocenti** inizializza()
{
    ListaDocenti **l;
    int i;

    l=(ListaDocenti**)malloc(sizeof(ListaDocenti*)*4);
    for(i=0; i<4; i++)
        l[i]=NULL;

    return l;
}

void stampa(ListaDocenti **l)
{
    stampaFile("reputazione.txt", l[3], "Alto");
    stampaFile("reputazione.txt", l[2], "Medio");
    stampaFile("reputazione.txt", l[1], "Basso");
    stampaFile("licenziabili.txt", l[0], NULL);
}

int main()
{
    Docente d;
    int reputazione;
    ListaDocenti **l; //uso un vettore di liste
    FILE *f;

    l=inizializza();

    f=fopen("docenti.txt","r");
    while(!feof(f))
    {
        d=leggiDocente(f);
        reputazione=decidiReputazione(d);
        l[reputazione]=CONS(l[reputazione],d);
    }
    fclose(f);

    stampa(l);
    //aggiungere funzioni di deallocazione del vettore di liste
    return 0;
}

```