

**Fondamenti di Informatica L-A - Ing. Gestionale/Ing. dei Processi Gestionali**  
**Prof. Penzo**  
**Soluzione compito del 17/01/2009**

**Esercizio 1**

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

#define MAX_UTENTI 10000
#define MAX_PRESTITI 50000

typedef enum{false, true} boolean;

typedef struct{
    char ISBN[10];
    char autore[20];
    char titolo[30];
    char editore[20];
    int anno;
    boolean inPrestito;
} libro;

typedef struct{
    int codiceUtente;
    char nome[15];
    char cognome[15];
    int eta;
} utente;

typedef struct{
    int gg,mm,aa;
}date;

typedef struct{
    int codicePrestito;
    int codiceUtente;
    char ISBN[10];
    date dataPrestito;
} prestito;

// prototipi delle funzioni
void contaUtentiGiovanissimi(utente U[], int numUtenti, int *utentiGiovanissimi);
boolean disponibilitaLibro(char filename[], char codISBN[]);
void mostraLibriPerAutore(char filename[], char autore[]);
int aggiornaPrestiti(prestito P[], int numPrestiti, prestito p, char filename[]);

void main(){

    utente UTENTI[MAX_UTENTI];
    int numUtenti; // variabile contenente il numero corrente di utenti nel vettore
    int giovanissimi;
```

```

// punto a)
contaUtentiGiovanissimi(UTENTI,numUtenti,&giovanissimi);
printf("Il numero di utenti giovanissimi e': %d\n", giovanissimi);

}

```

```

// punto 1)
void contaUtentiGiovanissimi(utente U[], int numUtenti, int *utentiGiovanissimi){

*utentiGiovanissimi = 0;
int i;

for (i=0; i<numUtenti; i++)
    if(U[i].eta < 14)
        (*utentiGiovanissimi)++;
}

```

```

// punto 2)
boolean disponibilitaLibro(char filename[], char codISBN[]){

FILE *fp;
boolean trovato=false, disponibile=false;
libro l;

fp = fopen(filename,"rb");
if (fp==NULL)
    printf("Errore! impossibile aprire il file\n");
else{
    fread(&l,sizeof(libro),1,fp);
    while(!feof(fp) && !trovato){
        if(!strcmp(l.ISBN,codISBN)){
            trovato=true;
            if(!l.inPrestito)
                disponibile=true;
        }
        fread(&l,sizeof(libro),1,fp);
    }
}
return(disponibile);
}

```

```

// punto 3
void mostraLibriPerAutore(char filename[], char autore[]){

FILE *fp;
boolean trovato=false;
libro l;

fp = fopen(filename,"rb");
if (fp==NULL)

```

```

        printf("Errore! impossibile aprire il file\n");
else{
    fread(&l,sizeof(libro),1,fp);
    while(!feof(fp)){
        if(!strcmp(l.autore,autore)){
            trovato=true;
            printf("Codice ISBN: %s\tTitolo: %s\n",l.ISBN,l.titolo);
        }
        fread(&l,sizeof(libro),1,fp);
    }
}
if (!trovato)
    printf("Errore! Autore non trovato\n");
}

```

**// punto 4)**

```

int aggiornaPrestiti(prestito P[], int numPrestiti, prestito p, char filename[]){

```

```

    boolean disponibile;

```

```

    disponibile = disponibilitaLibro(filename, p.ISBN);

```

```

    if(disponibile){

```

```

        if(numPrestiti < MAX_PRESTITI){ // c'e' ancora spazio nel vettore

```

```

            P[numPrestiti].codicePrestito = p.codicePrestito;

```

```

            P[numPrestiti].codiceUtente = p.codiceUtente;

```

```

            strcpy(P[numPrestiti].ISBN, p.ISBN);

```

```

            P[numPrestiti].dataPrestito = p.dataPrestito;

```

```

            numPrestiti++;

```

```

        }

```

```

        else

```

```

            printf("Impossibile inserire il prestito: vettore pieno\n");

```

```

    }

```

```

    else

```

```

        printf("Impossibile inserire il prestito: libro non disponibile\n");

```

```

    return(numPrestiti);

```

```

}

```