Corso di Informatica Medica

Esercitazione IX

Alessandro A. Nacci
nacci@elet.polimi.it - alessandronacci.com

Converti in decimale...

1010

Converti in decimale...

10110

22

Converti in decimale...

10110

22 ... numero di giorni all'esame!

Converti in decimale...

1000010000

Converti in decimale...

1000010000

528

Converti in decimale...

1000010000

528 ... numero di ORE all'esame!

Co nale...

100 000

52 di ORE

!

READY...
SET...
GO

Automobili: terza versione

QUALCHE LEZIONE FA..

- Rappresentare in C una automobile. Nel nostro caso, una automobile è descritta da un nome, un costo, un colore, da un insieme di componenti e da un libretto di circolazione.
- Un componente ha un nome, un costo ed una categoria. Le categorie possibili sono TRAZIONE, MULTIMEDIA, SICUREZZA
- Il libretto di circolazione riporta invece l'anno e la provincia di immatricolazione e in che classe Euro rientra.

- Il programma deve poter permettere la creazione di auto e la stampa a schermo di tutti i dati relativi ad un'auto
- Deve poter permettere inoltre di modificare il nome dell'auto
- Deve poter calcolare il costo totale per la produzione dell'auto

Automobili: Le strutture dati - Codice C

QUALCHE LEZIONE FA..

- AUTOMOBILE
  - NOME
  - COSTO
  - COLORE
  - COMPONENTI
    - NOME
    - COSTO
    - CATEGORIA
  - LIBRETTO
    - ANNO IMM.
    - PROVINCIA
    - CLASSE EURO

```c
typedef enum {TRAZIONE, MULTIMEDIA,
              SICUREZZA} tipi_categoria;

typedef struct {
    int anno_immatricolazione;
    char provincia[STR_LEN];
    int classe_euro;
} libretto_circolazione;

typedef struct {
    char nome[STR_LEN];
    float costo;
    tipi_categoria categoria;
} componente;

typedef struct {
    char nome[STR_LEN];
    float costo;
    char colore[STR_LEN];
    int numero_componenti;
    componente* componenti;
    libretto_circolazione libretto;
} automobile;
```

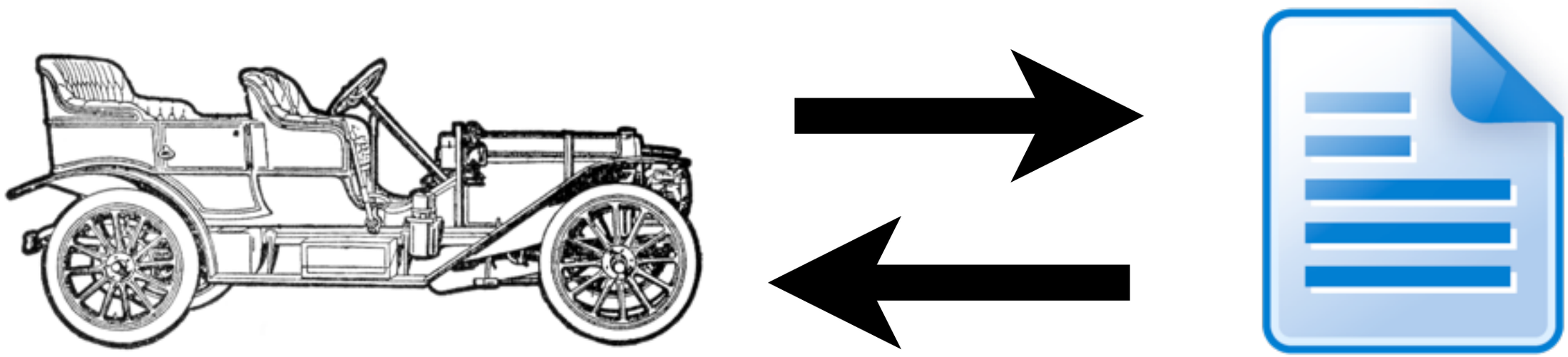
```c
automobile crea_auto(char nome[STR_LEN], double costo, char colore[STR_LEN],
            int numero_componenti, componente* componenti,
            libretto_circolazione libretto)
{

   printf("Creo una nuova autovettura di nome: %s\n", nome);

   automobile autovettura;

   strcpy(autovettura.nome,nome);
   autovettura.costo = costo;
   strcpy(autovettura.colore,colore);
   autovettura.numero_componenti = numero_componenti;
   autovettura.componenti = componenti;
   autovettura.libretto = libretto;

   return autovettura;

}
```

```c
libretto_circolazione crea_libretto_circolazione(int anno_immatricolazione,
                 char provincia[STR_LEN], int classe_euro)
{

   libretto_circolazione libretto;
   libretto.anno_immatricolazione = anno_immatricolazione;
   strcpy(libretto.provincia,provincia);
   libretto.classe_euro = classe_euro;

   return libretto;
}
```

```c
componente crea_componente(char nome[STR_LEN],
                 double costo, int categoria)
{
   componente c;

   strcpy(c.nome, nome);
   c.costo = costo;
   c.categoria = categoria;

   return c;
}
```

Lavoriamo con i file

- Vogliamo poter salvare tutte le informazioni di una automobile su file e poterle rileggere indietro



Per poter salvare un'automobile...

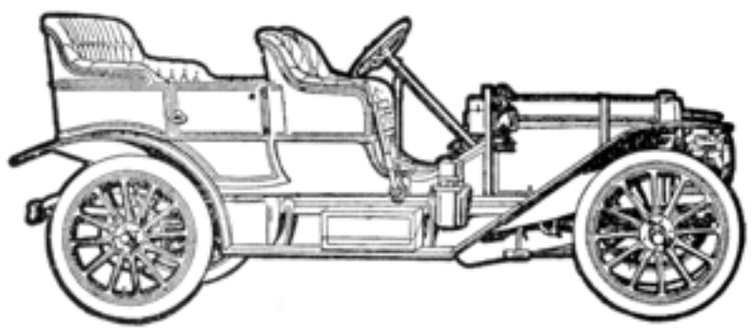
- AUTOMOBILE
  - NOME
  - COSTO
  - COLORE
  - COMPONENTI
    - NOME
    - COSTO
    - CATEGORIA
  - LIBRETTO
    - ANNO IMM.
    - PROVINCIA
    - CLASSE EURO

E' UN DATO STRUTTURATO!

DOBBIAMO POTER SALVARE I COMPONENTI

DOBBIAMO POTER SALVARE I LIBRETTI

Organizzazione dei dati

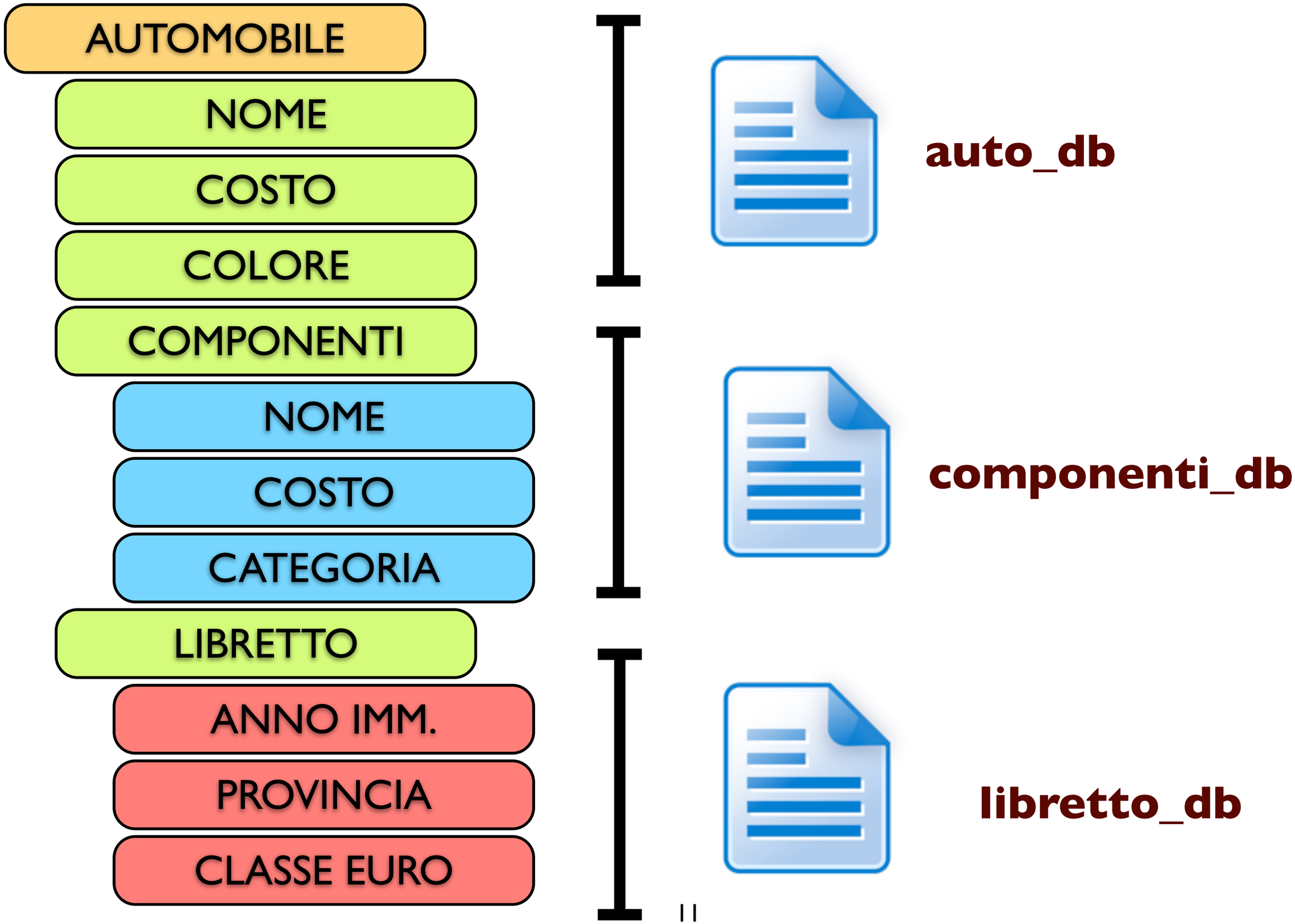


auto_db

componenti_db

libretto_db

Organizzazione dei dati



Scrittura su file

- Ci è comodo avere una funzione per la scrittura di una singola linea di un file...

```
void scrivi_linea_su_file(char linea[MAX_FILE_LINE], char* nome_file, char* mode)
{

}
```

Scrittura su file

- Ci è comodo avere una funzione per la scrittura di una singola linea di un file...

```
void scrivi_linea_su_file(char linea[MAX_FILE_LINE], char* nome_file, char* mode)
{
    FILE *fp_mio_file;

    fp_mio_file = fopen (nome_file, mode);

    if (fp_mio_file==NULL)
        printf("Si e' verificato un errore nell'apertura del file.\n");
    else
        printf("File aperto correttamente.\n");


    fprintf(fp_mio_file,"%s",linea);

    if (fclose (fp_mio_file)==0)
        printf("File chiuso correttamente.\n");
}
```

Salviamo un libretto...

```
typedef struct {
    int anno_immatricolazione;
    char provincia[STR_LEN];
    int classe_euro;
} libretto_circolazione;
```

```
void salva_libretto_circolazione(char* nome_auto, libretto_circolazione libretto)
{

}
```

Salviamo un libretto...

```
typedef struct {
    int anno_immatricolazione;
    char provincia[STR_LEN];
    int classe_euro;
} libretto_circolazione;
```

```
void salva_libretto_circolazione(char* nome_auto, libretto_circolazione libretto)
{
    char linea[MAX_FILE_LINE];
    sprintf(linea, "%s\t%d\t%s\t%d\n", nome_auto,
            libretto.anno_immatricolazione, libretto.provincia,
            libretto.classe_euro);
    scrivi_linea_su_file(linea, "libretto_db", "w");
}
```

Salviamo un libr

libretto_db

```
FIAT_BRAVO	2010	COMO	5
```

```c
typedef struct {
    int anno_immatricolazione;
    char provincia[STR_LEN];
    int classe_euro;
} libretto_circolazione;
```

```c
void salva_libretto_circolazione(char* nome_auto, libretto_circolazione libretto)
{
    char linea[MAX_FILE_LINE];
    sprintf(linea, "%s\t%d\t%s\t%d\n", nome_auto,
            libretto.anno_immatricolazione, libretto.provincia,
            libretto.classe_euro);
    scrivi_linea_su_file(linea, "libretto_db", "w");
}
```

Salviamo i componenti...

```
typedef struct {
    char nome[STR_LEN];
    float costo;
    tipi_categoria categoria;
} componente;
```

```
void salva_componenti(char* nome_auto, componente* componenti, int numero_componenti)
{

}
```

Salviamo i componenti...

```
typedef struct {
    char nome[STR_LEN];
    float costo;
    tipi_categoria categoria;
} componente;
```

```
void salva_componenti(char* nome_auto, componente* componenti, int numero_componenti)
{
    int i;
    char linea[MAX_FILE_LINE];

    for (i = 0; i < numero_componenti; i++)
    {
        sprintf(linea,"%s\t%s\t%f\t%s\n", nome_auto, componenti[i].nome,
            componenti[i].costo, stringa_categoria(componenti[i].categoria));
        scrivi_linea_su_file(linea, "componenti_db", "a");
    }
}
```

Salviamo i componenti...

```c
typedef struct {
    char nome[STR_LEN];
    float costo;
    tipi_categoria categoria;
} componente;
```

componenti_db — Edited

```
FIAT_PUNTO      FRENO   420.200012      TRAZIONE
FIAT_PUNTO      RUOTA   656.400024      TRAZIONE
```

```c
void salva_componenti(char* nome_auto, componente* componenti, int numero_componenti)
{
    int i;
    char linea[MAX_FILE_LINE];

    for (i = 0; i < numero_componenti; i++)
    {
        sprintf(linea,"%s\t%s\t%f\t%s\n", nome_auto, componenti[i].nome,
            componenti[i].costo, stringa_categoria(componenti[i].categoria));
        scrivi_linea_su_file(linea, "componenti_db", "a");
    }
}
```

Salviamo l'auto...

```
typedef struct {
    char nome[STR_LEN];
    float costo;
    char colore[STR_LEN];
    int numero_componenti;
    componente* componenti;
    libretto_circolazione librett
} automobile;
```

```
void salva_auto(automobile autovettura)
{

}
```

Salviamo l'auto...

```
typedef struct {
    char nome[STR_LEN];
    float costo;
    char colore[STR_LEN];
    int numero_componenti;
    componente* componenti;
    libretto_circolazione librett
} automobile;
```

```
void salva_auto(automobile autovettura)
{
    char linea[MAX_FILE_LINE];
    sprintf(linea, "%s\t%s\t%f\t%d\n", autovettura.nome, autovettura.colore,
        autovettura.costo, autovettura.numero_componenti );
    scrivi_linea_su_file(linea, "auto_db", "w");
    salva_componenti(autovettura.nome, autovettura.componenti,
        autovettura.numero_componenti);
    salva_libretto_circolazione(autovettura.nome, autovettura.libretto);
}
```

Salviamo l'auto...

auto_db — Edited

```
FIAT_BRAVO    BLU  2000.000000   2
```

```c
typedef struct {
    char nome[STR_LEN];
    float costo;
    char colore[STR_LEN];
    int numero_componenti;
    componente* componenti;
    libretto_circolazione librett
} automobile;
```

```c
void salva_auto(automobile autovettura)
{
    char linea[MAX_FILE_LINE];
    sprintf(linea, "%s\t%s\t%f\t%d\n", autovettura.nome, autovettura.colore,
        autovettura.costo, autovettura.numero_componenti );
    scrivi_linea_su_file(linea, "auto_db", "w");
    salva_componenti(autovettura.nome, autovettura.componenti,
        autovettura.numero_componenti);
    salva_libretto_circolazione(autovettura.nome, autovettura.libretto);
}
```

Scriviamo ora il codice per leggere i dati da file..

Lettura del libretto di circolazione

```
libretto_circolazione carica_libretto()
{

}
```

```
FIAT_BRAVO      2010    COMO    5
```

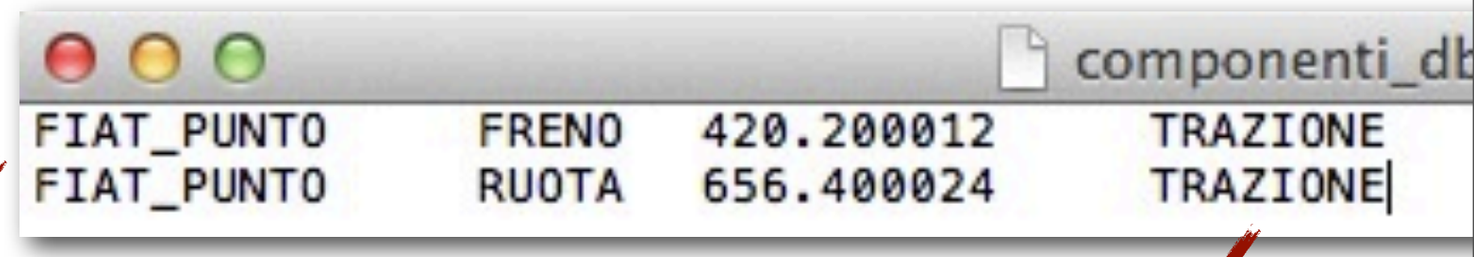
Lettura del libretto di circolazione

```
FIAT_BRAVO      2010    COMO    5
```

```c
libretto_circolazione carica_libretto()
{
    FILE *fp_mio_file;
    char nome_auto[STR_LEN];
    int anno_immatricolazione;
    char provincia[STR_LEN];
    int classe_euro;


    fp_mio_file = fopen ("libretto_db", "r");
    if (fp_mio_file==NULL)
        printf("Errore apertura file!\n");
    else{
        fscanf(fp_mio_file, "%s", nome_auto);
        fscanf(fp_mio_file, "%d", &anno_immatricolazione);
        fscanf(fp_mio_file, "%s", provincia);
        fscanf(fp_mio_file, "%d", &classe_euro);

    }

    return crea_libretto_circolazione(anno_immatricolazione, provincia, classe_euro);

}
```

Leggiamo i componenti

```
void carica_componenti(int numero_componenti, componente* componenti)
{
```



Sono più linee!

E' una stringa!

```
}
```

Leggiamo i componenti

```
void carica_componenti(int numero_componenti, componente* componenti)
{
```

componenti_db

```
FIAT_PUNTO      FRENO   420.200012      TRAZIONE
FIAT_PUNTO      RUOTA   656.400024      TRAZIONE
```

Sono più linee!

E' una stringa!

```
tipi_categoria categoria_stringa(char* categoria_str)
{

}
```

```
}
```

Leggiamo i componenti

```
void carica_componenti(int numero_componenti, componente* componenti)
{
```

componenti_db

```
FIAT_PUNTO      FRENO   420.200012      TRAZIONE
FIAT_PUNTO      RUOTA   656.400024      TRAZIONE
```

Sono più linee!

E' una stringa!

```
tipi_categoria categoria_stringa(char* categoria_str)
{


    if (strcmp(categoria_str, "TRAZIONE")) return TRAZIONE;
    if (strcmp(categoria_str, "MULTIMEDIA")) return MULTIMEDIA;
    if (strcmp(categoria_str, "SICUREZZA")) return SICUREZZA;

    printf("Errore conversione categoria! - %s\n", categoria_str);

    return 0;
}
```

```
}
```

Leggiamo i componenti

```
void carica_componenti(int numero_componenti, componente* componenti)
{
```

componenti_db

```
FIAT_PUNTO      FRENO    420.200012      TRAZIONE
FIAT_PUNTO      RUOTA    656.400024      TRAZIONE
```

Sono più linee!

E' una stringa!

```
tipi_categoria categoria_stringa(char* categoria_str)
{


    if (strcmp(categoria_str, "TRAZIONE")) return TRAZIONE;
    if (strcmp(categoria_str, "MULTIMEDIA")) return MULTIMEDIA;
    if (strcmp(categoria_str, "SICUREZZA")) return SICUREZZA;

    printf("Errore conversione categoria! - %s\n", categoria_str);

    return 0;
}
```

```
}
```

Leggiamo i componenti

```
void carica_componenti(int numero_componenti, componente* componenti)
{

    FILE *fp_mio_file;


    char nome_auto[STR_LEN];
    char nome[STR_LEN];
    float costo;
    char categoria_str[STR_LEN];
    tipi_categoria categoria;
    int i;

    printf("numero c:%d\n", numero_componenti);

    fp_mio_file = fopen ("componenti_db", "r");
    if (fp_mio_file==NULL)
        printf("Errore apertura file!\n");
    else{

        for (i = 0; i < numero_componenti; i++)
        {
            fscanf(fp_mio_file, "%s", nome_auto);
            fscanf(fp_mio_file, "%s", nome);
            fscanf(fp_mio_file, "%f", &costo);
            fscanf(fp_mio_file, "%s", categoria_str);
            categoria = categoria_stringa(categoria_str);


            componenti[i] = crea_componente(nome, costo, categoria);
        }
    }

    fclose(fp_mio_file);


}
```

componenti_db

```
FIAT_PUNTO      FRENO   420.200012      TRAZIONE
FIAT_PUNTO      RUOTA   656.400024      TRAZIONE
```

Leggiamo l'automobile

```
automobile carica_automobile(componente* componenti)
{

}
```

auto_db — Edited

```
FIAT_BRAVO     BLU  2000.000000   2
```

Leggiamo l'automobile

```
automobile carica_automobile(componente* componenti)
{
    FILE *fp_mio_file;
    char nome[STR_LEN];
    char colore[STR_LEN];
    float costo;
    int numero_componenti;
    libretto_circolazione libretto;

    fp_mio_file = fopen ("auto_db", "r");
    if (fp_mio_file==NULL)
        printf("Errore apertura file!\n");
    else{
        fscanf(fp_mio_file, "%s", nome);
        fscanf(fp_mio_file, "%s", colore);
        fscanf(fp_mio_file, "%f", &costo);
        fscanf(fp_mio_file, "%d", &numero_componenti);

    }

    fclose(fp_mio_file);
    carica_componenti(numero_componenti, componenti);
    libretto = carica_libretto();

    automobile autovettura = crea_auto(nome, costo, colore,
        numero_componenti, componenti, libretto);

    return autovettura;

}
```

auto_db — Edited

```
FIAT_BRAVO    BLU  2000.000000   2
```

Finiamo il main() ...

```
int main () {

    automobile autovettura;
    automobile autovettura_da_file;
    componente componenti[MAX_COMP];
    libretto_circolazione libretto;


    // Creiamo il componente "FRENO"
    componenti[0] = crea_componente("FRENO", 420.20, TRAZIONE);
    componenti[1] = crea_componente("RUOTA", 656.40, TRAZIONE);

    // Creiamo le informazioni del libretto
    libretto = crea_libretto_circolazione(2010, "COMO", 5);

    // Creiamo una autovettura
    autovettura = crea_auto("FIAT_BRAVO", 2000.00, "BLU", 2, componenti, libretto);

    // Stampiamo quello che abbiamo creato
    printf("\nBenvenuto!\n\n\n");
    stampa_auto(autovettura);

    salva_auto(autovettura);

    printf("\n\n\n\n");

    autovettura_da_file = carica_automobile(componenti);
    stampa_auto(autovettura_da_file);

    return 0;
}
```

Potete lasciare il vostro giudizio qui:

http://tinyurl.com/IEIMExe2013

Tutte il materiale sarà disponibile sul mio sito internet:

alessandronacci.com