



IEIM 2019-2020

Esercitazione III *“Stringhe e Struct”*

Alessandro A. Nacci

alessandro.nacci@polimi.it - www.alessandronacci.it



Esercizio I (a)

- Quale è l'output del seguente codice?

```
#include <stdio.h>

int main(){

    int arr[3] = {100,200,300};

    printf("%d\n", arr[0]);
    printf("%d\n", arr[1]);
    printf("%d\n", arr[2]);
    printf("%d\n", arr[3]);
    return 0;
}
```



Esercizio I (a)

- Quale è l'output del seguente codice?

```
#include <stdio.h>

int main(){

    int arr[3] = {100,200,300};

    printf("%d\n", arr[0]);
    printf("%d\n", arr[1]);
    printf("%d\n", arr[2]);
    printf("%d\n", arr[3]);
    return 0;
}
```

```
Terminal — bash — 80x24
> ./es1_1
> 100
> 200
> 300
> 1564782
```



Esercizio I (a)

- Quale è l'output del seguente codice?

```
#include <stdio.h>

int main(){

    int arr[3] = {100,200,300};

    printf("%d\n", arr[0]);
    printf("%d\n", arr[1]);
    printf("%d\n", arr[2]);
    printf("%d\n", arr[3]);
    return 0;
}
```

```
Terminal — bash — 80x24
> ./es1_1
> 100
> 200
> 300
> 1564782
Valori "a caso"!
```



Esercizio I (b)

- Quale è l'output del seguente codice?

```
#include <stdio.h>

int main(){

    int arr[3];

    printf("%d\n", arr[0]);
    printf("%d\n", arr[1]);
    printf("%d\n", arr[2]);

    arr[0] = 100;
    arr[1] = 200;
    arr[2] = 300;

    printf("%d\n", arr[0]);
    printf("%d\n", arr[1]);
    printf("%d\n", arr[2]);

    return 0;
}
```



Esercizio I (b)

- Quale è l'output del seguente codice?

```
#include <stdio.h>

int main(){

    int arr[3];

    printf("%d\n", arr[0]);
    printf("%d\n", arr[1]);
    printf("%d\n", arr[2]);

    arr[0] = 100;
    arr[1] = 200;
    arr[2] = 300;

    printf("%d\n", arr[0]);
    printf("%d\n", arr[1]);
    printf("%d\n", arr[2]);

    return 0;
}
```

```
Terminal — bash — 80x24
> ./es1_2
> 534534
> 324823
> 346234
> 100
> 200
> 300
```



Esercizio I (b)

- Quale è l'output del seguente codice?

```
#include <stdio.h>

int main(){

    int arr[3];

    printf("%d\n", arr[0]);
    printf("%d\n", arr[1]);
    printf("%d\n", arr[2]);

    arr[0] = 100;
    arr[1] = 200;
    arr[2] = 300;

    printf("%d\n", arr[0]);
    printf("%d\n", arr[1]);
    printf("%d\n", arr[2]);

    return 0;
}
```

```
Terminal — bash — 80x24
> ./es1_2
> 534534
> 324823
> 346234
> 100
> 200
> 300
```

Valori "a caso"!



ES2 - Schedine: problema

- Il programma, partendo da un array di dimensione prefissata che contiene in ogni cella:
 - data della giocata
 - colonna con i tredici segni
- Copia in un altro vettore solo le giocate dell'anno 2010
- Quindi, visualizza queste giocate.



ES2 - Schedine: dati

```
#include <stdio.h>

#define NUMSEGNI 13
#define NUMGIOCATE 3
#define ANNOCERCATO 2010

int main(){

    typedef struct {
        int giorno;
        int mese;
        int anno;
    } TipoData;

    typedef struct {
        TipoData data;
        char colonna[NUMSEGNI];
    } Giocata;

    Giocata giocate[NUMGIOCATE], trovate[NUMGIOCATE];
    int i, k, j = 0;
```



ES2 - Schedine: inserimento dati

```
Giocata giocate[NUMGIOCATE]
```

```
typedef struct {  
    TipoData data;  
    char colonna[NUMSEGNI];  
} Giocata;
```

```
typedef struct {  
    int giorno;  
    int mese;  
    int anno;  
} TipoData;
```

```
/* Inizio inserimento dati */  
for (i = 0; i < NUMGIOCATE; i++){  
    printf ("Giorno: ");  
    scanf ("%d", &giocate[i].data.giorno);  
    printf ("Mese: ");  
    scanf ("%d", &giocate[i].data.mese);  
    printf ("Anno: ");  
    scanf ("%d", &giocate[i].data.anno);  
    printf ("Colonna: \n");  
    /* Si inseriscono tutti i NUMSEGNI valori */  
    for (k = 0; k < NUMSEGNI; k++){  
        printf ("Valore parita-%d (1,x,2): ", k+1);  
        do{  
            scanf ("\n%c", &giocate[i].colonna[k]);  
        }while(!((giocate[i].colonna[k]=='1') || (giocate[i].colonna[k]=='x') || (giocate[i].colonna[k]=='2')));  
    }  
}  
/* Termine inserimento dati */
```



ES3 - Schedine: copia valori

```
/* Ricerca giocate fatte nell'anno ANNOCERCATO */  
for (i = 0; i < NUMGIOCATE; i++)  
    if (giocate[i].data.anno == ANNOCERCATO) {  
        /* Copia la giocata */  
        trovate[j] = giocate[i];  
        j++;  
    }
```



ES3 - Schedine: stamp dei risultati

```
/* Stampa a video del risultato */
for (i = 0; i < j; i++) {
    printf ("Data: %d/%d/%d\n", trovate[i].data.giorno,trovate[i].data.mese,trovate[i].data.anno);

    for (k = 0; k < NUMSEGNI; k++)
        printf ("%c ", trovate[i].colonna[k]);

    printf ("\n");
}

return 0;
}
```



Esercizio 3: stringhe - vocali consecutive

Scrivere un programma che data una stringa di lunghezza massima 100 stampi le occorrenze di coppie di vocali identiche consecutive.



Esercizio 3: stringhe - vocali consecutive

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

#define MAXLEN 100

int main(){

    char stringa[MAXLEN];
    int len, i, cont = 0;

    int vocale; //conterrà 0 o 1

    printf("inserire una stringa\n");
    scanf("%s", stringa);
    len = strlen(stringa);

    for(i=0; i<len-1; i++){

        vocale = 0;
        if(stringa[i] == 'a' || stringa[i] == 'e' ||
            stringa[i] == 'i' || stringa[i] == 'o' ||
            stringa[i] == 'u'){
            vocale = 1;
        }

        if(vocale == 1 && stringa[i+1] == stringa[i]){
            cont++;
            printf("la vocale ripetuta e' %c\nil numero di coppie e' %d\n", stringa[i], cont);
        }

    }

    if(cont == 0)
        printf("non ci sono vocali ripetute consecutivamente\n");

    return 0;

}
```



Esercizio 3: stringhe & array

Scrivere un programma che prenda stringhe di lunghezza massima 100 (quando l'utente inserisce la stringa "stop" il programma smette di chiedere inserimenti)

Calcolare quante stringhe di lunghezza L (definita dall'utente), $L+1$ e $L+2$ sono state inserite



Esercizio 3: stringhe & array

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

#define MAXLEN 100

int main(){

    char stringa[100];
    int l, len, i, count[3]={0,0,0};

    printf("inserire la lunghezza desiderata\n");
    scanf("%d", &l);

    do{

        printf("inserire una stringa\ninserire stop per terminare\n");
        scanf("%s", stringa);
        len = strlen(stringa);

        if(len >= l && len<=l+2 && strcmp(stringa, "stop") != 0){
            printf("%d\n", len-1);
            count[len-1]++;
        }

    }while(strcmp(stringa, "stop") != 0);

    for(i=0; i<3; i++){
        printf("il numero di parole di lunghezza %d inserite e': %d\n", l+i, count[i]);
    }

    system("PAUSE");
    return 0;

}
```



Utilizzo di Struct Complesse

“Gestione Automobili”



Automobili

- 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

```
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;
```



- Scriviamo una funzione per la creazione di un generico libretto di circolazione



- Scriviamo una funzione per la creazione di un generico libretto di circolazione

```
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;  
}
```



- Scriviamo una funzione per la creazione di un generico componente di un'auto



- Scriviamo una funzione per la creazione di un generico componente di un'auto

```
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;
}
```



- Scriviamo una funzione per la creazione di una generica automobile

- Scriviamo una funzione per la creazione di una generica automobile

```
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;
}
```



Stampa a schermo dei dati di un'auto

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO



Stampa di un componente

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO



Stampa di un componente

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

stringa

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO



Stampa di un componente

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

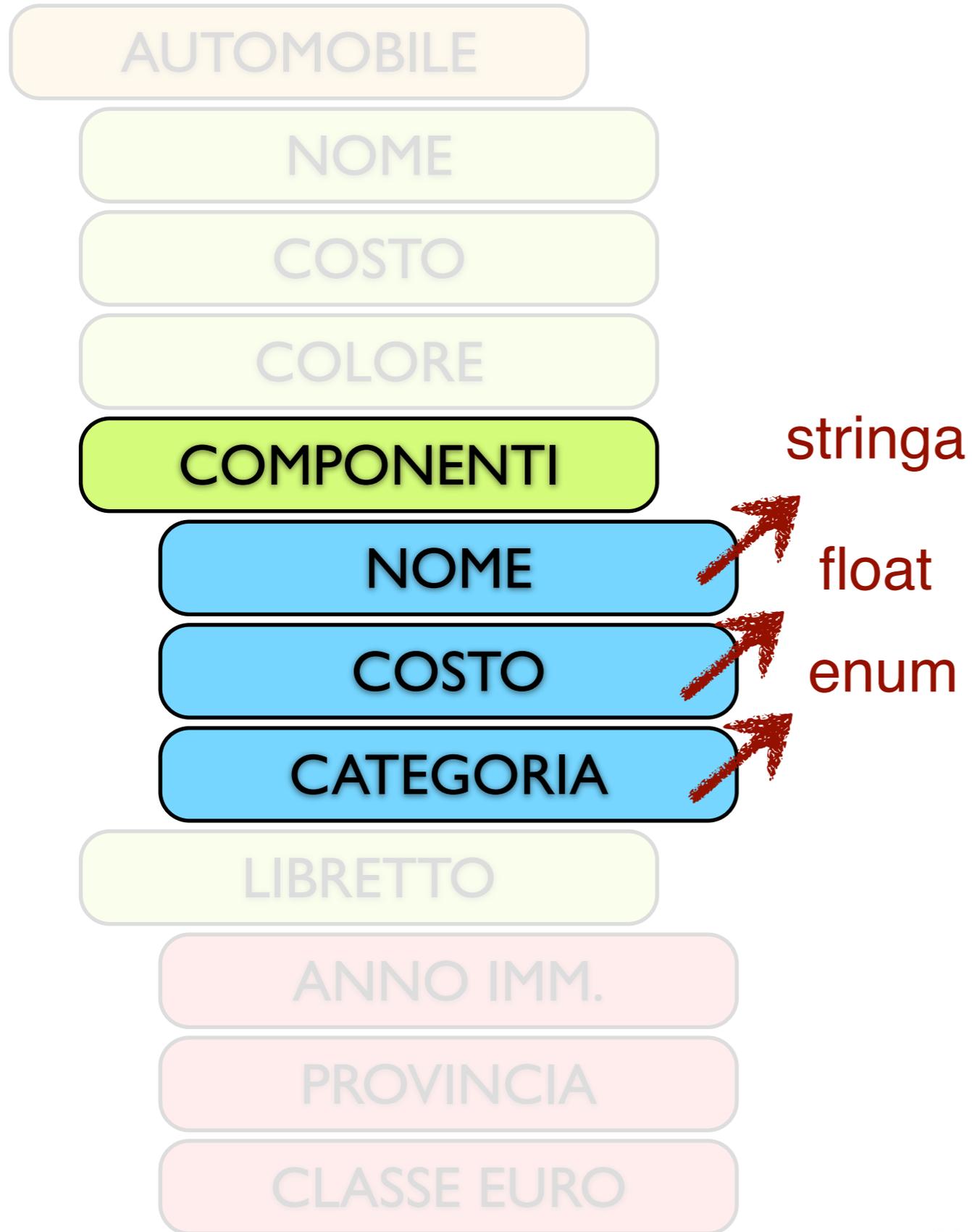
CLASSE EURO

stringa

float



Stampa di un componente





Stampa di un componente

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO

stringa

float

enum

```
char* stringa_categoria(tipi_categoria categoria)
{
    if (categoria == TRAZIONE) return "TRAZIONE";
    if (categoria == MULTIMEDIA) return "MULTIMEDIA";
    if (categoria == SICUREZZA) return "SICUREZZA";

    return "SCONOSCIUTO";
}
```



Stampa di un componente

```
void stampa_componenti(componente* componenti, int numero_componenti)
{
    int i;

    for (i = 0; i < numero_componenti; i++)
    {
        printf("Nome componente: %s |\t Costo: eur. %f |\t Categoria: %s \n",
            componenti[i].nome, componenti[i].costo,
            stringa_categoria(componenti[i].categoria));
    }
}
```

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO

stringa

float

enum

```
char* stringa_categoria(tipi_categoria categoria)
{
    if (categoria == TRAZIONE) return "TRAZIONE";
    if (categoria == MULTIMEDIA) return "MULTIMEDIA";
    if (categoria == SICUREZZA) return "SICUREZZA";

    return "SCONOSCIUTO";
}
```



Calcolo costo componente

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO

```
float calcola_costo_componenti(automobile autovettura)
{
    int i;
    float tot = 0;

    for (i = 0; i < autovettura.numero_componenti; i++)
    {
        tot += autovettura.componenti[i].costo;
    }

    return tot;
}
```



Stampa libretto circolazione

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO



Stampa libretto circolazione

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO

intero





Stampa libretto circolazione

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO

intero

stringa





Stampa libretto circolazione

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO

intero

stringa

intero



Stampa libretto circolazione

AUTOMOBILE

NOME

```
void stampa_libretto_circolazione(libretto_circolazione libretto)
{
    printf("Anno: %d |\t Prov.:%s |\t Euro:%d\n", libretto.anno_immatricolazione,
        libretto.provincia, libretto.classe_euro);
}
```

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO

intero

stringa

intero



Stampa a schermo dei dati di un'auto

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO



Stampa a schermo dei dati di un'auto

AUTOMOBILE

NOME

COSTO

```
void stampa_auto(automobile autovettura)
{
    printf("Nome: %s\n", autovettura.nome);
    printf("Colore: %s\n", autovettura.colore);
    printf("Costo: eur. %f\n", autovettura.costo);
    printf("Costo componenti: eur. %f \n", calcola_costo_componenti(autovettura) );
    printf("\nCOMPONENTI:\n");
    printf("-----\n");
    stampa_componenti(autovettura.componenti, autovettura.numero_componenti);
    printf("\nLIBRETTO CIRCOLAZIONE:\n");
    printf("-----\n");
    stampa_libretto_circolazione(autovettura.libretto);
}
```

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO



Una prima parte di main() ...

```
int main () {

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

    automobile* ptr0;
    automobile* ptr1;
    automobile* ptr2;
    automobile* ptr3;

    // 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);
    ptr0 = &autovettura;

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

    return 0;|
}
```



Occupiamoci della modifica

```
automobile modifica_nome_auto(automobile autovettura, char nuovo_nome[STR_LEN])  
{  
  
}
```

```
void modifica_nome_auto2(automobile autovettura, char nuovo_nome[STR_LEN])  
{  
  
}
```

```
void modifica_nome_auto3(automobile* autovettura, char nuovo_nome[STR_LEN])  
{  
  
}
```



Occupiamoci della modifica

```
automobile modifica_nome_auto(automobile autovettura, char nuovo_nome[STR_LEN])  
{  
    strcpy(autovettura.nome, nuovo_nome);  
    return autovettura;  
}
```

```
void modifica_nome_auto2(automobile autovettura, char nuovo_nome[STR_LEN])  
{  
  
}
```

```
void modifica_nome_auto3(automobile* autovettura, char nuovo_nome[STR_LEN])  
{  
  
}
```



Occupiamoci della modifica

```
automobile modifica_nome_auto(automobile autovettura, char nuovo_nome[STR_LEN])
{
    strcpy(autovettura.nome, nuovo_nome);
    return autovettura;
}
```

```
void modifica_nome_auto2(automobile autovettura, char nuovo_nome[STR_LEN])
{
    strcpy(autovettura.nome, nuovo_nome);
}
```

```
void modifica_nome_auto3(automobile* autovettura, char nuovo_nome[STR_LEN])
{

}
```



Occupiamoci della modifica

```
automobile modifica_nome_auto(automobile autovettura, char nuovo_nome[STR_LEN])
{
    strcpy(autovettura.nome, nuovo_nome);
    return autovettura;
}
```

```
void modifica_nome_auto2(automobile autovettura, char nuovo_nome[STR_LEN])
{
    strcpy(autovettura.nome, nuovo_nome);
}
```

```
void modifica_nome_auto3(automobile* autovettura, char nuovo_nome[STR_LEN])
{
    strcpy(autovettura->nome, nuovo_nome);
}
```



Occupiamoci della modifica

```
automobile modifica_nome_auto(automobile autovettura, char nuovo_nome[STR_LEN])
{
    strcpy(autovettura.nome, nuovo_nome);
    return autovettura;
}
```

```
void modifica_nome_auto2(automobile autovettura, char nuovo_nome[STR_LEN])
{
    strcpy(autovettura.nome, nuovo_nome);
}
```

```
void modifica_nome_auto3(automobile* autovettura, char nuovo_nome[STR_LEN])
{
    strcpy(autovettura->nome, nuovo_nome);
}
```



Finiamo il main() ...

```
int main () {

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

    automobile* ptr0;
    automobile* ptr1;
    automobile* ptr2;
    automobile* ptr3;

    // 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);
    ptr0 = &autovettura;

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

    printf("\nModifico nome auto...\n\n\n");
    autovettura = modifica_nome_auto(autovettura, "FIAT PUNTO");
    stampa_auto(autovettura);

    printf("\nModifico nome auto...\n\n\n");
    modifica_nome_auto2(autovettura, "FIAT ULISSE");
    stampa_auto(autovettura);

    printf("\nModifico nome auto...\n\n\n");
    modifica_nome_auto3(&autovettura, "FIAT PANDA");
    stampa_auto(autovettura);

    return 0;
}
```



Automobili: Le strutture dati - Codice C

RICAPITOLIAMO

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO

```
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;
```



```
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;
}

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;
}

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;
}
```



Note



Note



Note



Note



Note



Note



Note



Note



Note



Note

**Tutte il materiale sarà
disponibile sul mio sito
internet!**

www.alessandronacci.it

See You Next Time!

