IEIM 2017-2018

Esercitazione XII

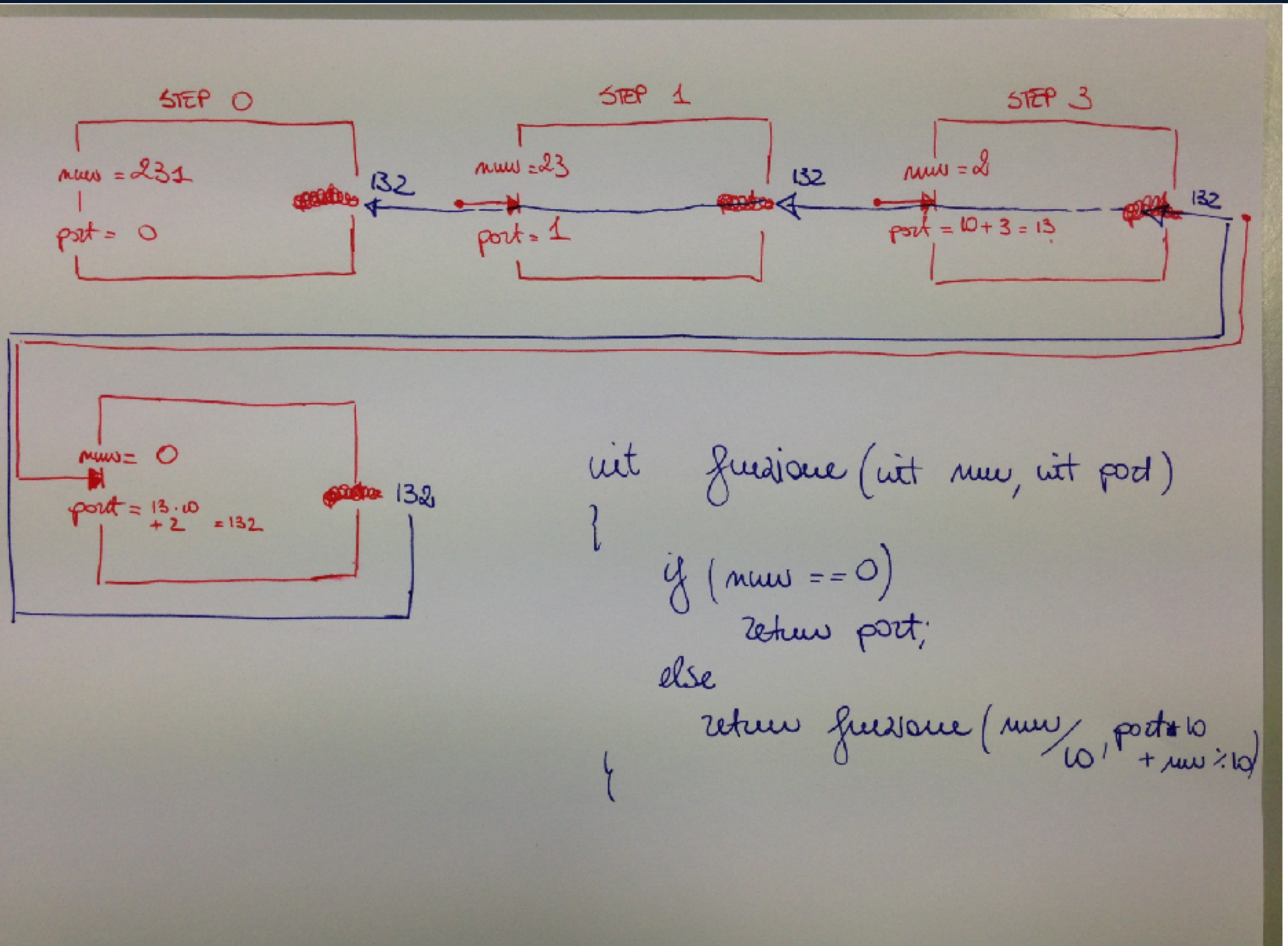
"Ripasso"

Alessandro A. Nacci
alessandro.nacci@polimi.it - www.alessandronacci.it

Ricorsione - Esercizio III

- Analizzare il comportamento della seguente funzione ricorsiva mostrando l'andamento delle variabili, considerando i seguenti input:
  - *funzione(231,0)*
- Dire quale funzione espleta

```
int mia_funzione(int num, int part) {
      if (num == 0)
            return part;
      else {
            return mia_funzione(num/10, part*10 + num%10);
      }
}
```




```
int funzione (int num, int port)
{
    if (num == 0)
        return port;
    else
        return funzione (num/10, port*10 + num%10)
{
```

Ordinamento di vettori

“Il lancio dei dadi”

Lancio dei dadi

Data la sequenza di lanci di un dato che ha prodotto il seguente risultato:

{3,4,6,1,3,5,4,2,6,2,3}

- Calcolare media e mediana degli elementi di un array contenente i lanci di un dado.
- Calcolare inoltre le frequenze di ogni valore.

La media

Dato che sappiamo usare le sommatorie... $M_a = \frac{1}{n}\sum_{i=1}^{n} x_i$

```
//Funzione che calcola il valore medio di un array
void calcolaMedia(int array[], int *media){
```

```
}
```

Cosa sarebbe cambiato se fosse stato...

```
int calcolaMedia(int array[])
```

La mediana

- Per calcolare la mediana di dati:
  - si ordinano gli n di dati in ordine crescente o decrescente;

    Come facciamo ad ordinare i dati?
  - se il numero di dati è dispari la mediana corrisponde al valore centrale, ovvero al valore che occupa la posizione $(n + 1) / 2$
  - se il numero n di dati è pari, la mediana è stimata utilizzando i due valori che occupano le posizione $(n / 2)$ e $((n/2) + 1)$ (generalmente si sceglie la loro media

Ordinamento dei dati: bubble sort

- Per avere un ordinamento crescente:
  - scambio man mano a due a due due elementi x_i e x_{i+1} se $x_i > x_{i+1}$
- Dopo n-1 iteraizoni ho la garanzia di aver ordinato l'intero array

```
void swap(int *a, int *b)
{
     int tmp;
     tmp = *a;
     *a = *b;
     *b = tmp;
}
```

```
void bubbleSort(int arr[]){

     int i,j;

     for ( i = 1 ; i < N ; i++ )
          for ( j = 0 ; j < N - i ; j++ )
          {
                    if (arr[j] > arr[j + 1])
                      swap(&arr[j], &arr[j+1]);


          }

}
```

La mediana (codice C)

```c
//Funzione che calcola la mediana di un array
void calcolaMediana(int array[], int *mediana){

}
```

Calcolo delle frequenze

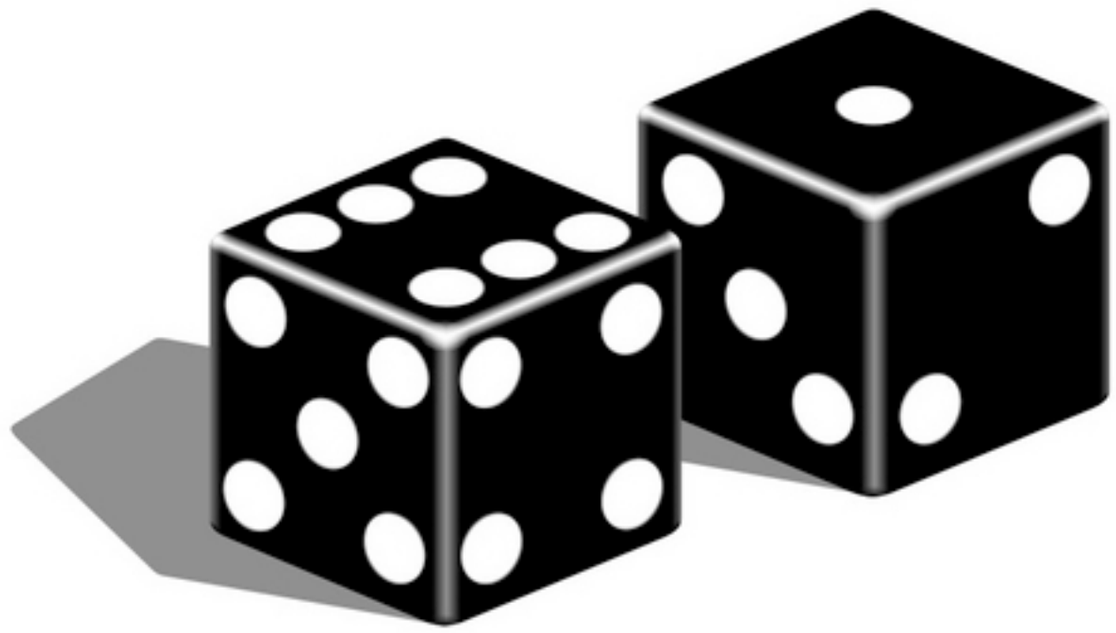
```
//Funzione che calcola le frequenze dei valori dei lanci
void frequenze(int array[], int freq[]){

}
```

Il main()

```c
int main(){

    int lanci[N]={3,4,6,1,3,5,4,2,6,2,3};

    //array contenente le frequenze dei valori da 0 a 5, rappresentanti i valori da 1 a 6 del dado
    int frequenze_array[6]={};
    int i;


    //Stampa dei lanci
    for(i=0;i<N;i++)
    {
        printf("%d ", lanci[i]);
    }

    int media=0, mediana;
    calcolaMedia(lanci, &media);
    calcolaMediana(lanci, &mediana);
    frequenze(lanci, frequenze_array);

    printf("\nLa media e' %d\n", media);
    printf("La mediana e' %d\n", mediana);
    for(i=0;i<6;i++)
    {
        printf("La frequenza di %d e' %d\n", i+1, frequenze_array[i]);
    }

    return 0;

}
```

FINE ESERCIZIO

Domande?



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

```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;
```

crea_libretto_cicolazione

- 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;
}
```

crea_componente

- Scriviamo una funziona 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;
}
```

crea_auto

- Scriviamo una funziona 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

Stampa di un componente

AUTOMOBILE

NOME

COSTO

COLORE

COMPONENTI

NOME

COSTO

CATEGORIA

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO

```
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));
   }
}
```

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.

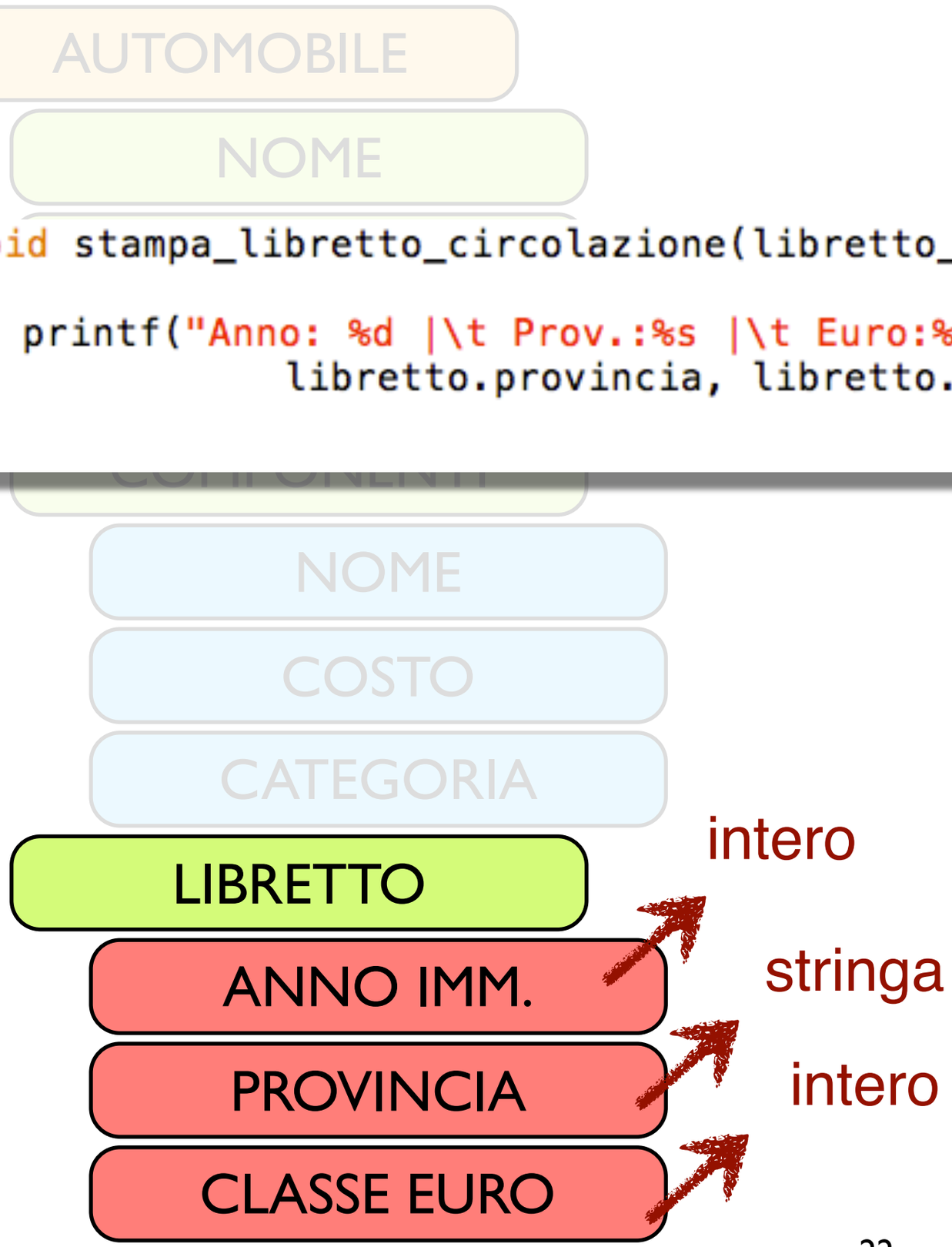
PROVINCIA

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

```
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);
}
```



Stampa a schermo dei dati di un'auto

AUTOMOBILE

NOME

COSTO

LIBRETTO

ANNO IMM.

PROVINCIA

CLASSE EURO

```
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);
}
```

Una prima parte di main() ...

```c
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])
{


}
```

Finiamo il main() ...

```c
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

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

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

```c
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;
}
```

Tutte il materiale sarà disponibile sul mio sito internet!

alessandronacci.it