UNIVERSITY OF TRENTO - Italy

Esercitazione 12

DISI – aa 2023/2024

Pierluigi Roberti
Carmelo Ferrante

Università degli Studi di Trento
pierluigi.roberti@unitn.it

Struct e funzioni

- Le funzioni possono prendere come argomento anche delle struct:

```
typedef struct Tdata{                  //Definizione della struttura Tdata
   int mese;
   int anno;
   int giorno;
} Tdata;

void stampaData(Tdata d);  //Dichiarazione del prototipo della funzione

int main(void) {
  Tdata d1;
  d1.anno = 2009;
  d1.mese = 3;
  d1.giorno = 10;
  stampaData(d1);          //Invocazione della funzione
}

void stampaData (Tdata d) { //Definizione della funzione "
  printf("%d/%d/%d", d.giorno, d.mese, d.anno);
}
```

Struct e funzioni

- Le funzioni possono restituire una struct :

```
typedef struct Tdata{
   int mese;
   int anno;
   int giorno;
} Tdata;
void stampaData(Tdata d);
Tdata inizializzaData(int g, int m, int a);  //Ritorna una struct

int main(void) {
  Tdata d1;
  d1 = inizializzaData(3,10,2009);
  stampaData(d1);
}
void stampaData (Tdata d) {
   printf("%d/%d/%d", d.giorno, d.mese, d.anno);
}
Tdata inizializzaData(int g, int m, int a){
   Tdata d;
   d.giorno = g;
   d.mese = m;
   d.anno = a;
   return d;
}
//NB: questo codice produce lo stesso risultato del precedente
```

Obiettivo dell'esercitazione

- Definire strutture e funzioni che ci permettano di realizzare un'agenda.
- Passi necessari:
  - Definire una struttura per rappresentare una data
  - Definire una struttura per rappresentare un evento.
  - Definire una struttura che rappresenta l'agenda
- Ogni struttura verrà manipolata tramite l'utilizzo di funzioni.

Esercizio 1 (1/2)

- Definire una struttura adatta a contenere una data. Nella data devono comparire:
  - anno
  - mese
  - giorno
  - ora
  - minuti
- Definire le seguenti funzioni che operano su tale struttura:
  - Una funzione che inizializza una struttura data con i valori specificati in argomento:

    ```
    Tdata inizializzaData(int a,int me,int g,int o,int mi);
    ```

  - Una funzione che stampa la data passata in argomento:

    ```
    void stampaData(Tdata d);
    ```

    Il formato richiesto per la stampa è il seguente:

    ```
    giorno/mese/anno ora:minuti
    ```

Esercizio 1 (2/2)

- Una funzione che prende come argomenti due variabili di tipo Tdata e ritorna i seguenti valori
  - -1 se la prima data è precedente alla seconda
  - 0 se le due date sono uguali
  - 1 altrimenti

```
int confrontaDate(Tdata d1, Tdata d2);
```

Soluzione Esercizio 1

```
typedef struct Tdata{
     int anno;
     int mese;
     int giorno;
     int ora;
     int minuti;
} Tdata;


Tdata inizializzaData(int anno, int mese, int giorno, int ora, int minuti){
    Tdata daRitornare;
    daRitornare.anno = anno;
    daRitornare.mese = mese;
    daRitornare.giorno = giorno;
    daRitornare.ora = ora;
    daRitornare.minuti = minuti;
    return daRitornare;
}


void stampaData(Tdata d){
        printf("%2d/%2d/%4d %02d:%02d",
                                    d.giorno,d.mese,d.anno,d.ora,d.minuti);
}
```

Soluzione Esercizio 1 ver 1

```
int confrontaDate(Tdata d1, Tdata d2)
{  if(d1.anno == d2.anno)
   { if(d1.mese == d2.mese)
     { if(d1.giorno == d2.giorno)
       { if(d1.ora == d2.ora)
         { if(d1.minuti == d2.minuti)
              return 0;
           if(d1.minuti < d2.minuti)
               return -1;
           return 1;
         }//end if(d1.ora == d2.ora)
         if(d1.ora < d2.ora)
            return -1;
         return 1;
       }//end if(d1.giorno == d1.giorno)
       if(d1.giorno < d2.giorno)
         return -1;
       return 1;
     }//end if(d1.mese == d2.mese)
     if(d1.mese < d2.mese)
       return -1;
     return 1;
   }//end if(d1.anno == d2.anno)
   if(d1.anno < d2.anno)
     return -1;
   return 1;
}
```

Soluzione Esercizio 1 ver 2

```
int confrontaDate(Tdata d1, Tdata d2){
  if (d1.anno<d2.anno) { return -1; }
  if (d1.anno>d2.anno) { return +1; }
  if (d1.mese<d2.mese) { return -1; }
  if (d1.mese>d2.mese) { return +1; }
  if (d1.giorno<d2.giorno) { return -1; }
  if (d1.giorno>d2.giorno) { return +1; }
  if (d1.ora<d2.ora) { return -1; }
  if (d1.ora>d2.ora) { return +1; }
  if (d1.minuti<d2.minuti) { return -1; }
  if (d1.minuti>d2.minuti) { return +1; }
  return 0;
}
```

Esercizio 2

- Definire la seguente enum:
  `typedef enum {LEZIONE, PISCINA, APPUNTAMENTO, PALLAVOLO, STUDIO} Tattivita;`
- Definire una struttura adatta a descrivere un evento (*Tevento*). Questi sono i campi che deve avere:
  - **inizio** di tipo Tdata
  - **fine** di tipo Tdata
  - **attivita** di tipo Tattivita

  Un evento potrebbe essere ad esempio: una lezione che inizia alle 8:30 del 12/05/2009 e termina alle 9:30 del 12/05/2009.
- Definire le seguenti funzioni che operano su tale struttura:
  - Una funzione che inizializza un evento con i valori specificati in argomento:
    ```
    Tevento inizializzaEvento(Tdata inizio, Tdata fine,
                              Tattivita attivita);
    ```
  - Una funzione che stampa una variabile di tipo attivita:
    ```
    void stampaAttivita(Tattivita a);
    ```
  - Una funzione che stampa un evento:
    ```
    void stampaEvento(Tevento  e);
    ```
    Questo deve essere il formato di stampa:

    `Inizio: Data Inizio`
    `Fine: Data Fine`
    `Tipo di attività`

    NB: Le date devono essere stampate con le funzioni precedentemente definite

Soluzione Esercizio 2

```
typedef enum {LEZIONE, PISCINA, APPUNTAMENTO, PALLAVOLO, STUDIO} Tattivita;

typedef struct {
   Tdata inizio;
   Tdata fine;
   Tattivita attivita;
} Tevento;

void stampaAttivita(Tattivita a){
    switch(a){
        case LEZIONE:
            printf("Lezione");
            break;
        case PISCINA:
            printf("Piscina");
            break;
        case APPUNTAMENTO:
            printf("Appuntamento");
            break;
        case PALLAVOLO:
            printf("Pallavolo");
            break;
        case STUDIO:
            printf("Studio");
            break;
    }
}
```

Soluzione Esercizio 2

```
Tevento inizializzaEvento(Tdata inizio, Tdata fine, Tattivita attivita){
   Tevento daRitornare;
   daRitornare.inizio = inizio;
   daRitornare.fine = fine;
   daRitornare.attivita = attivita;
   return daRitornare;
}

void stampaEvento(Tevento e){
   printf("Inizio: ");
   stampaData(e.inizio);
   printf("\n");
   printf("Fine: ");
   stampaData(e.fine);
   printf("\n");
   stampaAttivita(e.attivita);
}
```

Esercizio 3 (1/4)

- Lo scopo di questo esercizio è quello di ridefinire l'agenda estendendo il lavoro fatto sinora.
- Rappresenteremo l'agenda con una struttura che contiene due campi:
  - Un vettore di tipo *Tevento* che chiamiamo `eventi` di dimensione N costante
  - Un intero che ci serve per memorizzare il numero di eventi contenuti nell'agenda e che chiamiamo `n_eventi`. Questo intero rappresenta anche l'indice della prima posizione libera all'interno del vettore `eventi`.

Esercizio 3 (2/4)

- Agenda concreta:

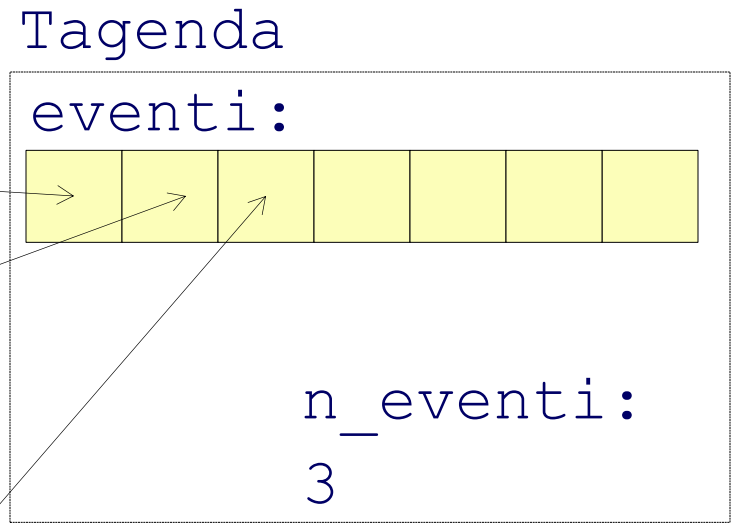
18 Maggio:

- 8:30 - 9:30
  Lezione
- 9:30 - 11:30
  Piscina

19 Maggio

- 14:00 - 13:30
  Appuntamento

- Astrazione:



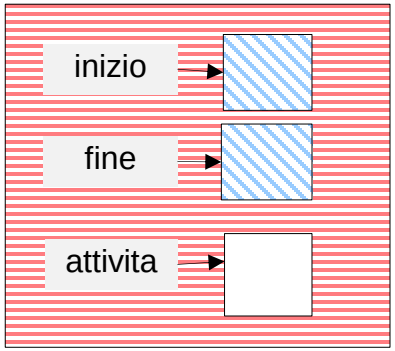
Esercizio 3 (3/4)

- Inizializzare un'agenda vuota:
  - Inizializzare un'agenda in modo che sia vuota significa associare ad `n_eventi` il valore zero. Così facendo si definisce che l'agenda non contiene nessun evento e che il prossimo evento aggiunto potrà essere salvato nella posizione zero all'interno del vettore `eventi`.
- Aggiungere un evento all'agenda:
  - Per aggiungere un elemento dobbiamo:
    - salvare il nuovo evento all'interno del vettore `eventi` nella posizione indicata da `n_eventi`.
    - aumentare di un'unità il valore di `n_eventi`.
  - Ad esempio se `n_eventi` prima dell'aggiunta del nuovo evento valeva 0, il nuovo evento verrà aggiunto in posizione zero all'interno del vettore `eventi` e la variabile `n_eventi` assumerà il valore 1. Notate che `n_eventi` rappresenta sempre il numero di eventi contenuti e la prima posizione libera all'interno del vettore.

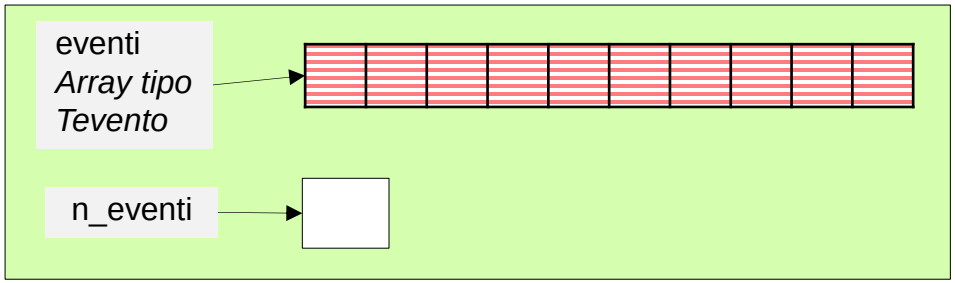
Strutture dati

Tdata

Tevento



Tagenda



Esercizio 3 (4/4)

- Stampare un'agenda:
  - La stampa di un'agenda prevede che tutti gli eventi presenti nel vettore `eventi` siano stampati come segue:

```
Evento in posizione 0:
Stampa dell'evento

Evento in posizione 1:
Stampa dell'evento
...
```

  - NB: gli eventi devono essere stampati con la funzione appositamente definita in precedenza
- Prototipi delle funzioni da definire:
  - `Tagenda inizializzaAgenda();`
  - `Tagenda aggiungiEvento(Tagenda ag, Tevento ev);`
  - `void stampaAgenda(Tagenda a);`

Struct Tagenda

```
typedef struct Tagenda{
    Tevento eventi [N_MAX_EVENTI];
    int n_eventi;
} Tagenda;
```

Array

Dimensione massima

Dimensione effettiva

Numero elementi in array

Indice prima posizione disponibile

Soluzione Esercizio 3

```
typedef struct Tagenda{
   Tevento eventi [N_MAX_EVENTI];
   int n_eventi;
} Tagenda;

Tagenda inizializzaAgenda(){
   Tagenda daRitornare;
   daRitornare.n_eventi = 0;
   return daRitornare;
}

Tagenda aggiungiEvento(Tagenda a, Tevento e){
   if (a.n_eventi >= N_MAX_EVENTI){
      printf("Errore, l'agennda è piena\n");
   } else {
      a.eventi[a.n_eventi] = e;
      a.n_eventi++;
   }
   return a;
}
```

Soluzione Esercizio 3

```
void stampaAgenda(Tagenda a){
        int i;
        for (i = 0; i < a.n_eventi; i++){
                printf("Evento in posizione %d:`n", i);
                stampaEvento(a.eventi[i]);
                printf("\n\n");
        }
}
```

Main per testare le funzioni

```
int main(){
        Tdata d1, d2;
        Tevento e;
        Tagenda a;

        a = inizializzaAgenda();

        d1 = inizializzaData(2009,6,10,10,30);
        d2 = inizializzaData(2009,6,10,11,30);

        e = inizializzaEvento(d1,d2,PISCINA);
        a = aggiungiEvento(a, e);

        d1 = inizializzaData(2009,6,11,14,30);
        d2 = inizializzaData(2009,6,11,16,0);

        e = inizializzaEvento(d1,d2,STUDIO);
        a = aggiungiEvento(a, e);

        d1 = inizializzaData(2009,7,11,16,0);
        d2 = inizializzaData(2009,6,11,16,30);

        e = inizializzaEvento(d1,d2,APPUNTAMENTO);
        a = aggiungiEvento(a, e);

        stampaAgenda(a);
        system("PAUSE");
        return 0;
}
```