



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Fondamenti di Informatica T

Orari (Refactoring)

## Esercizio: Orari (Refactoring)

**Si vuole ristrutturare il codice di uno degli esercizi già visti**

- Si scarichi dal sito del corso lo start kit per questa lezione
- Si consideri il codice della sottocartella "orari"
- Lo si redistribuisca in questo modo:
  - orari.h: definizioni di tipi di dato e dichiarazioni (no main)
  - orari.c: definizione delle funzioni (no main)
  - main.c: funzione main

Si verifichi che il codice ristrutturato funzioni ancora





ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Fondamenti di Informatica T

Somma Orari (Refactoring)

## Esercizio: Somma Orari (Refactoring)

**Si vuole ristrutturare il codice di uno degli esercizi già visti**

- Si scarichi dal sito del corso lo start kit per questa lezione
- La cartella "somma\_orari" contiene le soluzioni di due esercizi
- Si riorganizzi il codice in "es\_orari" nel modo seguente:
  - orari.h: tipi di dato e dichiarazioni (no main)
  - orari.c: definizione delle funzioni (no main)
- Si riorganizzi il codice in "es\_somma\_orari" nel modo seguente:
  - somma\_orari.h: tipi di dato e dichiarazioni (no main)
  - somma\_orari.c: definizione delle funzioni (no main)



## Esercizio: Somma Orari (Refactoring)

**Si vuole ristrutturare il codice di uno degli esercizi già visti**

Si sviluppi un nuovo main nel file "main.c", che:

- Legga tre orari con la funzione "leggi\_orario"
- Li inserisca in modo ordinato in un array con "inserisci\_orario"
- Sommi gli orari nell'array con "somma\_orari"





ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Fondamenti di Informatica T

Prodotti (Refactoring)

# Prodotti (Refactoring)

**Si vuole ristrutturare il codice di uno degli esercizi già visti**

- Si scarichi dal sito del corso lo start kit per questa lezione
- Si consideri il codice della sottocartella "prodotti"
- Lo si redistribuisca in questo modo:
  - prodotti.h: definizioni di tipi di dato e dichiarazioni (no main)
  - prodotti.c: definizione delle funzioni (no main)
  - main.c: funzione main

Si verifichi che il codice ristrutturato funzioni ancora





ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

# Fondamenti di Informatica T

Punti e Segmenti (Refactoring)

## Esercizio: Punti e Segmenti (Refactoring)

**Si vuole ristrutturare il codice di uno degli esercizi già visti**

- Si scarichi dal sito del corso lo start kit per questa lezione
- La cartella "punti" contiene le soluzioni di due esercizi
- Si riorganizzi il codice in "es\_punti" nel modo seguente:
  - orari.h: tipi di dato e dichiarazioni (no main)
  - orari.c: definizione delle funzioni (no main)
- Si riorganizzi il codice in "es\_segmenti" nel modo seguente:
  - somma\_orari.h: tipi di dato e dichiarazioni (no main)
  - somma\_orari.c: definizione delle funzioni (no main)
  - main.c: il main

Si verifichi che il codice ristrutturato funzioni ancora





ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

## Fondamenti di Informatica T

Andrea Acquaviva <andrea.acquaviva@unibo.it>

Michele Lombardi <michele.lombardi2@unibo.it>

Andrea Borghesi <andrea.borghesi3@unibo.it>

Giuseppe Tagliavini <giuseppe.tagliavini@unibo.it>

Allegra De Filippo <allegra.defilippo@unibo.it>