

Fondamenti di Informatica
Ingegneria Clinica
Lezione 15/10/2009



Prof. Raffaele Nicolussi
FUB - Fondazione Ugo Bordoni
Via B. Castiglione 59 - 00142 Roma



Docente	Raffaele Nicolussi	rnicolussi@fub.it 0654803323
Lezioni Aula 54 (ex aula 4) Via del Castro Laurenziano, 7	Lunedì, Giovedì, Venerdì	12:00 – 13:30
Esercitazioni Aula 15 Via Tiburtina, 205	Giovedì	14:00 – ???
Ricevimento:	Per appuntamento	in FUB, per email, per telefono
Sito web:	http://w3.uniroma1.it/IngClinFondinf	



Lezioni ed esami

□ Primo ciclo

- dal 12 Ottobre 2009 al 18 Dicembre 2009

□ Esami:

- due appelli
 - Dal 10 gennaio al 12 febbraio 2009
(verificare)
- due appelli di recupero
 - da definire

Materiale didattico



- Harvey M. Deitel, Paul J. Deitel. C. Corso completo di programmazione. Apogeo, 2007 (3° edizione)

- Materiale integrativo (dal sito web)
 - Dispensa per l'uso del compilatore IDE DevC++

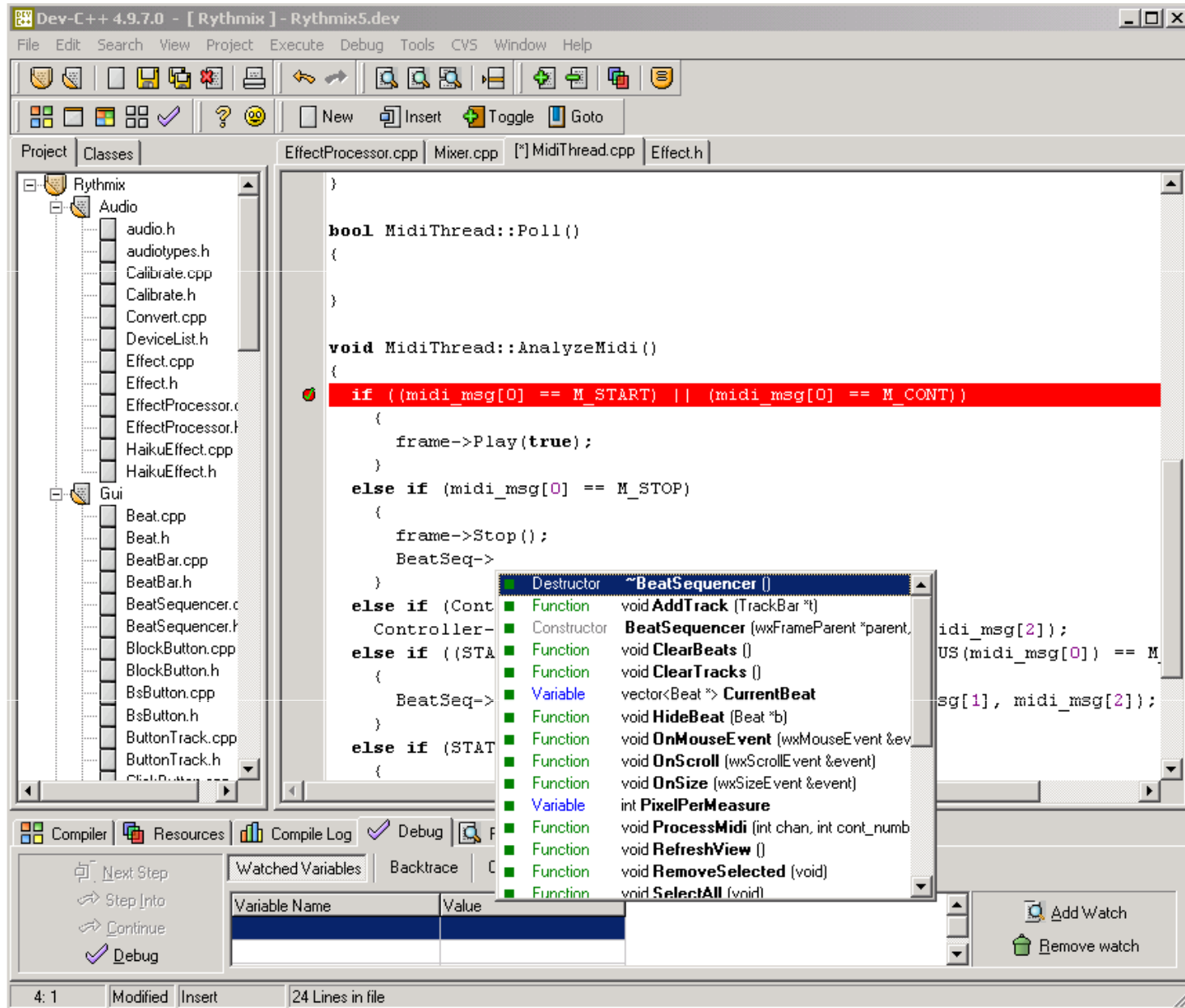
- Compilatori
 - IDE Dev C++ (v. 5 beta) <http://www.bloodshed.net>
 - lcc-win32 <http://www.cs.virginia.edu/~lcc-win32/>

- Altro materiale sarà messo a disposizione sul sito predisposto per il corso



C Corso completo di programmazione 3a edizione







 A screenshot of the LCC-win32 IDE interface. The main window shows a multi-tabbed editor with three open files: 'event.c', 'system.c', and 'http.c'. The 'http.c' file is currently selected and displays the following C code:


```

#include <stdio.h>
#include "main.h"

int running_standalone() {
    if (getenv("SERVER_SOFTWARE"))
        if (getenv("SERVER_NAME"))
            if (getenv("GATEWAY_INTERFACE"))
                if (getenv("REQUEST_METHOD"))
                    return 0;
            else
                return 1;
        else
            return 1;
    else
        return 1;
}

void output(char *msg) {
    /*check if header was written */
    if (! header_written) {
        printf("Content-type: text/html\n\n");
        header_written = 1;
    }
    /*write buffer */
    printf(msg);
}
    
```

 The IDE interface includes a menu bar (File, Edit, Search, Options, Project, Resources, Compiler, Utils, Versions, Window, Help) and a status bar at the bottom showing 'TOOCHARGER.COM' and 'running_standalone 6.1'.



Obiettivi

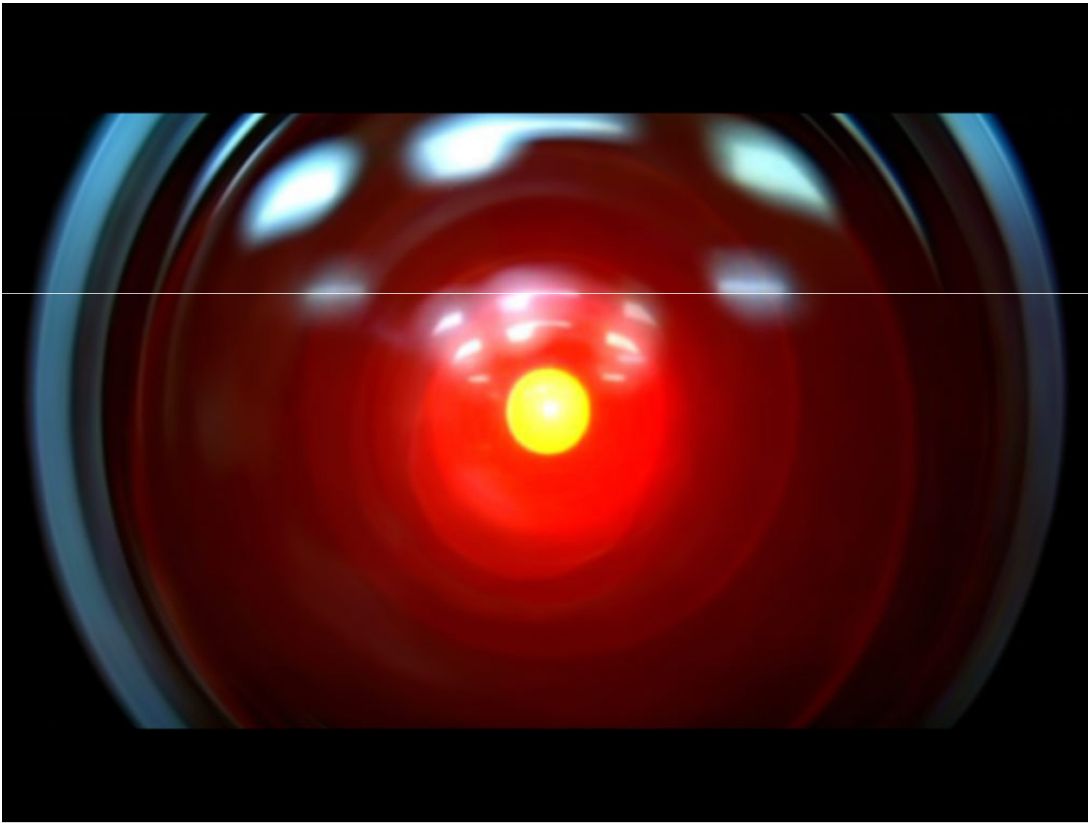
- **Imparare il concetto di algoritmo e programma**
- **Imparare a risolvere problemi in modo algoritmico**
 - **Imparare a tradurre l'algoritmo in programma**
- **Imparare a scrivere il programma in linguaggio C**
 - **Imparare alcuni algoritmi fondamentali**
 - **Teoria dell'immagine**



Cosa è un calcolatore

- ❑ Un calcolatore è una macchina che esegue gli ordini dati
- ❑ Gli ordini sono espressi sotto forma di istruzioni in un linguaggio di programmazione
- ❑ Le istruzioni vengono sintetizzate in un programma
- ❑ Il programma permette l'elaborazione di dati in ingresso per produrre dei dati in uscita

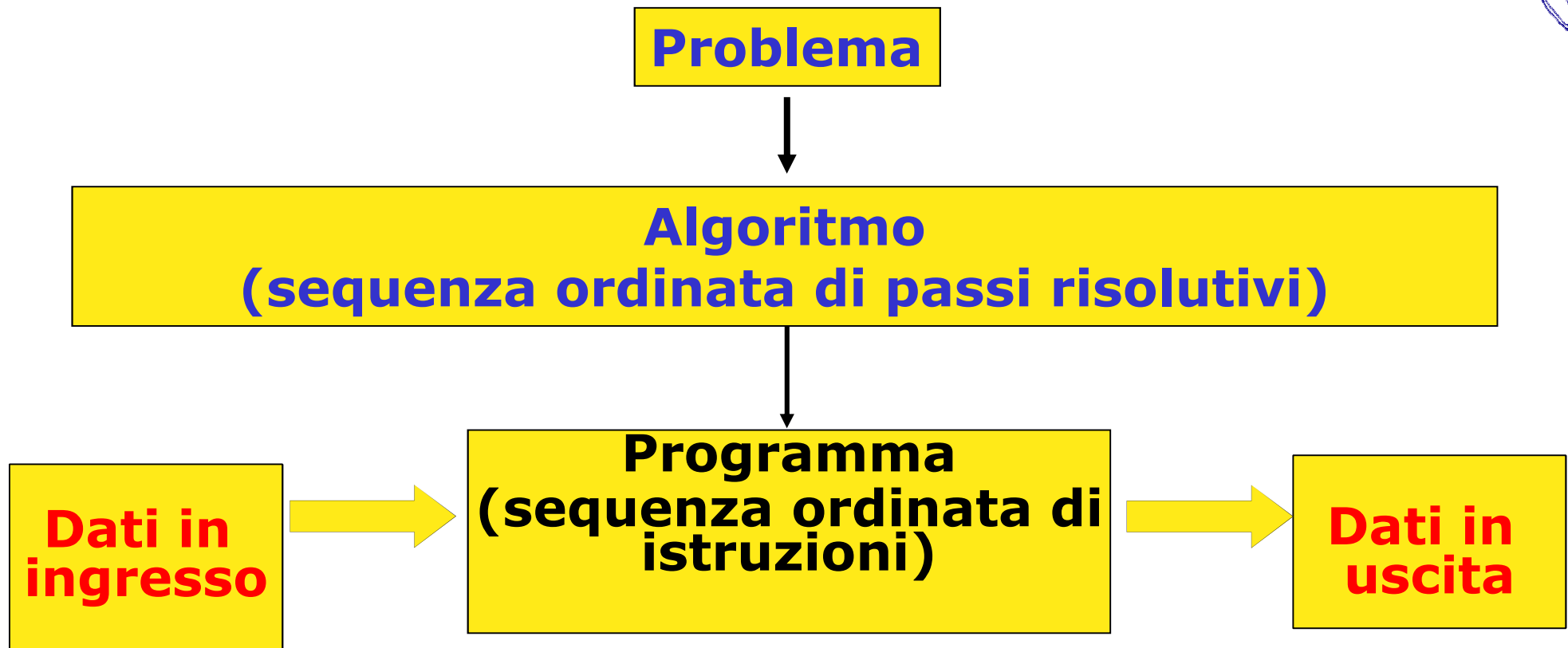
Calcolatore



- **HAL 9000** è il nome del computer di bordo della nave spaziale del film **2001: Odissea nello spazio** di **Stanley Kubrick**.
- L'origine del nome pare che sia attribuibile ad un gioco di parole con il nome della IBM, nota azienda costruttrice di hardware. Infatti applicando il **cifrario di Cesare** con uno spostamento di una lettera alle tre lettere **H-A-L** risultano le lettere **I-B-M**.



Risoluzione di un problema



Il programma è la traduzione in un linguaggio di programmazione del procedimento risolutivo (ossia dell'algoritmo)



Definizione formale di algoritmo

« Insieme di istruzioni elementari (univocamente interpretabili) che, eseguite in un ordine stabilito, permettono la soluzione di un problema in un numero finito di passi »



Algoritmo

- In **informatica**, con il termine **algoritmo** si intende un metodo per la soluzione di un problema adatto a essere implementato sotto forma di **programma**.
- Un **algoritmo** si può definire come un *procedimento* che consente di *ottenere* un *risultato* atteso eseguendo, in un determinato ordine, un insieme di *passi semplici* corrispondenti ad azioni scelte solitamente da un insieme finito



Algoritmo (2)

- Il termine deriva dal nome del matematico persiano **Muhammad ibn Mūsa 'l-Khwārizmī**, che si ritiene essere uno dei primi autori ad aver fatto riferimento esplicitamente a questo concetto, nel libro *Kitāb al-djabr wa 'l-muqābala* (*Libro sulla ricomposizione e sulla riduzione*),
- Dallo stesso libro prende anche le origini la parola **algebra**.
- Tuttavia gli algoritmi erano presenti anche nelle antiche tradizioni matematiche, ad esempio la **matematica babilonese**, quella cinese o del Kerala



Algoritmo (3)

- Nel senso più ampio della parola, "**algoritmo**" è anche una ricetta di cucina, o la sezione del libretto delle istruzioni di una lavatrice che spiega come programmare un lavaggio.
- Di norma, comunque, la parola viene usata in contesti matematici (fin dalle origini) e soprattutto informatici (più recentemente).
- Un esempio più appropriato di algoritmo potrebbe essere, quindi, il procedimento per il calcolo del massimo comune divisore o del minimo comune multiplo.



Esempio di Algoritmo: l'MCD

- In matematica, il massimo comun divisore (M.C.D.) di due numeri interi, che non siano entrambi uguali a zero, è il numero naturale più grande per il quale possono entrambi essere divisi.
 - Ad esempio:
 - $\text{MCD}(12, 18) = 6$
 - $\text{MCD}(-4, 14) = 2$
 - $\text{MCD}(5, 0) = 5$
- Il calcolo dell'MCD si svolge con l'Algoritmo di Euclide



Algoritmo di Euclide in linguaggio naturale

- Obiettivo: calcolare $\text{MCD}(a,b)$
- Dati due numeri naturali a e b , si controlla se b è zero.
 - Se lo è, a è il MCD.
 - Se non lo è
 - si divide a / b e si assegna ad r il resto della divisione (operazione indicata con " a modulo b " più sotto).
 - Se $r = 0$ allora si può terminare affermando che b è il MCD cercato
 - altrimenti occorre assegnare $a = b$ e $b = r$ e si ripete nuovamente la divisione.



Algoritmo di Euclide in linguaggio del calcolatore

```
function MCD(a, b)
  while a  $\neq$  b
    if a > b a := a - b
    else b := b - a
  return a
```



Esercizi

- Provate ad esprimere alcune attività quotidiane sotto forma di algoritmi in linguaggio naturale
 - Ricercare l'oggetto x (biancheria, ...) in un armadio di n cassetti
 - Fare una lavatrice
 - Usare un'automobile
 - ...
- Come implementereste una sorta di Cifrario di Cesare?
 - Es: data una parola \mathbf{P} in ingresso di n caratteri ottenere la parola \mathbf{C} (cifrata) in uscita in cui ogni lettera sia spostata in avanti di x caratteri



Semplice Cifrario di cesare

- Ci sono ancora lettere da leggere in **P**?
 - Leggo lettera **l**
 - Sposto lettera in avanti di x : $ll = l + x$
 - Scrivo **ll** in **C**
 - Mi sposto alla lettera successiva di **P**
- Stampo **C**