Laboratorio di Programmazione
&
Informatica 1

(Laurea triennale in matematica)

Laboratorio di Programmazione Strutturata

(Laurea triennale in Scienze e Tecnologie per i Media)

Lezione 1: Presentazione corso

Informazioni generali

- Docente:

  Prof.ssa Dora Giammarresi

  giammarr@mat.uniroma2.it

- Tutor:

  Daniele Fakhoury

Informazioni generali

- Orario lezioni :
  - Martedì 9:00 -11:00 Lab. aula 17
  - Martedì 14:00 -16:00 aula 2 (o 17)
  - Giovedì 9:00 -11:00 Lab. aula 17 (o 2)
  - Giovedì 11:00 -13:00 Lab. aula 17
- Numero crediti: 10

Finalità del corso

- Principi fondamentali della programmazione.
- Programmazione in C.
- Elementi di algoritmi e strutture dati.

Libri di testo

Harvey M. Deitel, Paul J. Deitel
Il linguaggio C

Fondamenti e Tecniche di Programmazione
Pearson Education.

Dispense (fornite in seguito) per la parte di

Algoritmi e strutture dati

Esami

- Prova laboratorio
  - scrivere programma in C sul computer
- Prova scritta
  - esercizi di programmazione in C (scritti su carta) + esercizi teorici
- Orale
  - discussione del compito
  - domande sulla parte teorica

Esami

- Quando?
  - Sessione estiva (giugno/luglio)
  - Sessione autunnale (settembre)
  - Sessione invernale (febbraio)

NOTA:

La prova di laboratorio e il compito scritto possono sostenuti in sessioni diverse (stesso A.A.). L'orale nella stessa sessione dello scritto.

Esami

- Cosa fare per passare l'esame?

- (studiare)
- Esercitarsi a programmare sul computer!

Pagina web del corso

www.mat.uniroma2.it/

~giammarr/Teaching/LabInf1/index.html

Introduzione al corso

“Non ho paura dei computer, ma della loro eventuale mancanza.” (Isaac Asimov)

- I **computer** sono onnipresenti nella nostra vita quotidiana (prima erano legati al solo calcolo scientifico).

la Repubblica.it

“L'informatica non riguarda i computer più di quanto l'astronomia riguardi i telescopi. ” (E. W.Dijkstra)

- L’ **informatica** è una disciplina scientifica e non coincide con le sue applicazioni.
- L’ **informatica** non consiste nello scrivere programmi per un computer!
- I **computer** sono uno strumento: l’informatico può lavorare anche solo con carta e penna!
- (anche se un Informatico dovrebbe saper programmare...)

"L'informatica è la scienza che consente di ordinare, trattare e trasmettere l'informazione attraverso l'elaborazione elettronica" (dizionario Devoto-Oli)

- La parola **informatica** deriva dalla parola francese, coniata nel 1962, *Informatique* come forma contratta di *Information Automatique*.

“L'informatica è la scienza degli algoritmi che descrivono e trasformano l’informazione : la loro teoria, analisi, progetto, efficienza, realizzazione e applicazione.”

(ACM: Association for Computing Machinery)

- **Algoritmo:** procedura di calcolo ben definita che riceve un insieme di valori in **input** e produce un corrispondente insieme di valori in **output**.
- Serve a risolvere un problema computazionale

"Computational thinking will be a fundamental skill used by everyone in the world by the middle of the 21st Century

(Jeannette M. Wing, 2006)

- Pensare come un informatico è molto di più che programmare un computer
- Il "Computational Thinking" ha a che fare con la risoluzione di problemi (problem solving): ogni problema si affronta, usando opportunamente l'astrazione e la decomposizione, ricorrendo alle tecniche sviluppate dall'informatica.
- Questo modo di pensare influenza altre discipline: biologia, neuroscienze, chimica, etc.

INFORMATICA

Algoritmo

DEFINIZIONE ALGEBRICA

Insieme completo delle regole che permettono la soluzione di un determinato problema

DEFINIZIONE OPERATIVA

Procedura effettiva che indica le istruzioni (passi) da eseguire per ottenere i risultati voluti a partire dai dati di cui si dispone

Algoritmi

- Utilizziamo algoritmi nella vita quotidiana tutte le volte che, ad es., seguiamo le istruzioni per il montaggio di una apparecchiatura, per impostare il ciclo di lavaggio di una lavastoviglie, per prelevare contante da uno sportello Bancomat, ecc.
- Un algoritmo è una sequenza di passi che, se intrapresa da un esecutore (non pensante!), permette di ottenere i risultati attesi a partire dai dati forniti.
- Una volta in grado di specificare un algoritmo per risolvere un problema, siamo anche in grado di automatizzare il procedimento descritto dall'algoritmo.

Da dove viene questa parola?

- Il termine algoritmo deriva dal nome del matematico persiano Muhammad ibn Musa al-Khwarizmi (Corasmia 780 circa - 850 circa). Esercitò la professione nella citt`a di Baghdad, dove insegnava, e introdusse nel mondo arabo i numeri indiani.
- La sua opera “Il calcolo degli indiani” venne successivamente tradotta in latino da un monaco europeo, con il titolo Liber algarismi - (Il libro di al-Khwarizmi).

Esempi di Algoritmi

Esempio Culinario:

• Ricetta x preparare la torta al cioccolato

Esempio Turistico:

• Indicazioni per raggiungere i Musei Vaticani

Esempi sui numeri:

• Insieme di passi per verificare se un numero è dispari, pari, primo, ecc.

Altri esempi (piu' difficili) :

• l'algoritmo di Euclide per il calcolo del MCD

Algoritmo di Euclide per MCD

DA WIKIPEDIA:

L'**algoritmo di Euclide** è un algoritmo per trovare il massimo comune divisore (MCD) tra due numeri interi.

È uno degli algoritmi più antichi conosciuti, essendo presente negli *Elementi* di Euclide[1] intorno al 300 a.C.; tuttavia, probabilmente l'algoritmo non è stato scoperto da Euclide, ma potrebbe essere stato conosciuto anche 200 anni prima.

Certamente era conosciuto da Eudosso di Cnido intorno al 375 a.C.; Aristotele (intorno al 330 a.C.) ne ha fatto cenno ne *I topici*, 158b, 29-35. L'algoritmo non richiede la fattorizzazione dei due interi.

Algoritmo di Euclide per MCD

Dati due numeri naturali *a* e *b*, si controlla se *b* è zero (*questa prima fase rientra ovviamente nell'ambito di un uso moderno dell'algoritmo ed era ignorata da Euclide e dai suoi predecessori, che non conoscevano lo 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 (**r=a modulo b**).

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** la divisione.

Algoritmo di Euclide per MCD
vers.1

1. r ← a mod b
2. a ← b
3. b ← r
4. *SE* b è diverso da 0 torna a 1.
5. MCD(a,b) ← a

NOTA: se b>a la prima iterazione scambia i valori di a e b

Algoritmo di Euclide per MCD vers.2

1. *Ripeti finchè* b è diverso da 0
2. r ← a mod b
3. a ← b
4. b ← r
5. MCD(a,b) ← a

NOTA: se b>a la prima iterazione scambia i valori di a e b

Algoritmo di Euclide per MCD
(vers. in linguaggio C)

```
int MCD ( int a, int b )
{
    int r;
    while (b!=0)
    {
       r = a%b;
       a = b;
       b = r;
    }
    return a;
}
```

Algoritmo: Caratteristiche

- Esprimibile con un numero finito di istruzioni
- Insieme di istruzioni di cardinalità finita
- Tempo di esecuzione di ogni istruzione finito
- Istruzioni eseguibili da un elaboratore (esecutore non pensante?)
- Non esiste limite alla lunghezza dei dati di ingresso
- Non c'e' un limite alla memoria disponibile
- Numero di passi esecuzione eventualmente illimitato

Algoritmo: caratteristiche (1)

- descrizione finita:
  - poter essere descritto da un numero finito di azioni elementari;
- l'esecutore opera in modo discreto:
  - l'esecutore esegue un passo alla volta, e ciascuna azione modifica lo stato della memoria. L'effetto di ciascun passo è univocamente determinato una volta noto lo stato dell'esecutore (cioe la memoria) che lo esegue.

Algoritmo: caratteristiche (2)

- illimitatezza dei dati in ingresso/uscita:
  - no ipotesi sulla dimensione massima dei dati in ingresso e sulla lunghezza dei risultati
- illimitatezza della memoria:
  - l'esecutore ha a disposizione una memoria illimitata per registrare dati intermedi durante l'esecuzione;
- illimitatezza dei passi eseguiti:
  - l'esecutore, in generale, esegue la sua procedura senza porre limiti al numero di passi eseguiti.

Problemi e Algoritmi e Programmi

- L'algoritmo è una descrizione precisa e non ambigua di una sequenza di passi da seguire per risolvere un problema. L'algoritmo e' quindi un raffinamento della soluzione. Gli algoritmi possono essere espressi in molti modi
- Il programma è una (possibile) descrizione dell'algoritmo espressa usando un particolare linguaggio di programmazione. Si intende che il programma debba essere completo ed eseguibile
- Spesso useremo direttamente il linguaggio C per descrivere gli algoritmi.

Il processo di programmazione

- Il processo di programmazione comprende tutte le attivita' necessarie per sviluppare dei programmi in modo che siano memorizzati e preparati per l'esecuzione

- Per sviluppare programmi su un computer è necessario un ambiente di sviluppo che almeno comprenda, oltre al sistema operativo, un editor ed un compilatore

Editor e compilatori

- Un programma è un testo che consiste in una sequenza di istruzioni scritte in un particolare linguaggio di programmazione. Tale "testo" viene creato e salvato come un file su disco tramite un editor.
  - Una volta che e' stato scritto e memorizzato, il (testo del) programma viene dato in input al compilatore

- Il compilatore ha una duplice funzione:

  – Controlla la validita' del programma: segnala gli errori di sintassi. Questi errori devono essere corretti tramite l'editor ed il programma corretto deve quindi essere nuovamente compilato

  – Se il programma non contiene errori, il compilatore traduce il programma in linguaggio macchina

Esecuzione dei programmi - 1

- Il compilatore naturalmente non è in grado di scoprire gli errori logici o di esecuzione (i bug) del programma: in tali casi, il programma risulta essere sintatticamente corretto ma non si comporta come ci si aspetta

- Il programma viene eseguito (o "fatto girare") attraverso un comando al sistema operativo che lo carica in memoria e quindi diventa un processo in esecuzione

- Gli errori logici possono essere prevenuti seguendo delle buone regole di programmazione

Esecuzione dei programmi - 2

- Spesso gli errori logici si individuano mandando in esecuzione il programma ed osservandone il comportamento tramite degli insiemi di test affidabili

- Una volta rilevato e corretto un errore logico, si dovrebbero nuovamente eseguire i test sul suo comportamento, in quanto le presunte "correzioni" potrebbero aver introdotto nuovi errori

- Il processo di rilevazione e correzione degli errori logici di un programma viene chiamato debugging

Iniziamo a programmare

- Cosa occorre:
- Carta e Penna

  (per preparare l'algoritmo)

- Un Editor di Testo
- Un compilatore

  (per codificare l'algoritmo)

} anche un IDE

Vediamo adesso l’algoritmo di Euclide «inserito» un programma in C.