Informazioni Generali

Marco Alberti


Dipartimento
di Matematica
e Informatica


Università
degli Studi
di Ferrara

Programmazione e Laboratorio, A.A. 2021-2022

Ultima modifica: 18 settembre 2021

Sommario

Sommario

Informazione

I dati sono informazione codificata per poter essere elaborata e scambiata.
L'informazione è la risposta (parziale) a una domanda; l'informazione riduce l'incertezza di chi la riceve.
L'informazione si misura in bit (binary digit) o multipli. Un bit è la quantità di informazione sufficiente a dimezzare l'incertezza (e ad annullarla se le scelte sono solo due).
Convenzionalmente i possibili valori di un bit sono 0 e 1.

Esempi

- *Quale* direzione prendere a un bivio (1 bit)
- *Quale* fra i numeri compresi fra 0 e 255 (8 bit = 1 byte)
- *Quale* fotografia (1 milione di byte)
- *Quale* contabilità annuale di un'azienda

Computer e informazione

I computer sono utili perché, programmati, aiutano a risolvere problemi di carenza di informazione.

Problema

Qual è la somma di due numeri? QUALI?

5 7 12

Il computer non crea informazione che non esiste; la rende immediatamente disponibile all'utente, elaborando quella fornita.

Problema

Che tempo fa adesso a Mosca?

Programmazione

Il corso ha l'obiettivo di introdurre le basi della programmazione dei calcolatori elettronici (computer) utilizzando il linguaggio di programmazione C.

- Computer: macchina in grado di
  - immagazzinare dati
  - elaborare dati
  - comunicare dati con l'esterno per mezzo di dispositivi di input (ingresso) e output (uscita)
- Le modalità di elaborazione e la comunicazione sono specificate da un programma, cioè una rappresentazione
  - delle informazioni gestite (strutture dati)
  - delle operazioni da eseguire (algoritmo)

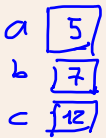
  in un linguaggio di programmazione.
- La programmazione è l'attività di produzione dei programmi.

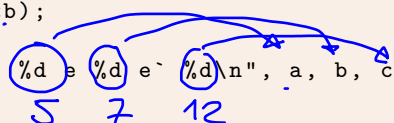
Un programma (in linguaggio C)

Università degli Studi di Ferrara

001_informazioni_generali/somma.c

```
1  #include <stdio.h>
2
3  int main() {
4    int a, b, c;
5    printf("Immetti due numeri interi:\n");
6    scanf("%d%d", &a, &b);
7    c = a + b;
8    printf("La somma di %d e %d e` %d\n", a, b, c);
9    return 0;
10 }
```





Quale problema risolve? FORNIRE LA SOMMA DI DUE NUMERI DATI

Contenuti del corso

- Concetti alla base della risoluzione di problemi per mezzo di calcolatore elettronico (hardware e software, macchina di Von Neumann e macchine astratte, input e output, algoritmo, programma, processo)
- Paradigma procedurale/imperativo di programmazione: espressioni e istruzioni, controllo di flusso, astrazione procedurale
- Tipi di dato (rappresentazione dell'informazione) primitivi, compositi e astratti
- Gestione della memoria
- Ricorsione
- Linguaggio di programmazione C

Capacità da acquisire

- Identificazione delle strutture dati (rappresentazione dell'informazione) e degli algoritmi (procedimenti di calcolo) adatti alla risoluzione di problemi per mezzo di un calcolatore
- Implementazione in linguaggio C di algoritmi e strutture dati
  (in particolare, operazioni su *sequenze* di dati primitivi, composti e astratti)
- Strutturazione modulare di un programma
- Utilizzo dei principali strumenti per la programmazione

Le nozioni saranno presentate in modo da favorire l'acquisizione progressiva di queste capacità, per problemi di complessità crescente.

Sommario

Attività didattiche - COVID-19 edition

010 pt 1
010 pt 2
020 pt 1
020 pt 2

Università degli Studi di Ferrara

ASSENTI

- Lezioni registrate: esposizione argomenti di teoria con esempi, demo e semplici esercizi
- Focus group (in presenza, streaming, registrazione): chiarimenti e approfondimenti (in presenza) MERCOLEDÌ
- Laboratorio (in presenza, streaming, registrazione): svolgimento guidato di esercizi più complessi GIOVEDÌ

GALLI

- Pre-corso: per chi ha difficoltà a seguire le prime lezioni. Calendario (online): `http://www.unife.it/scienze/informatica/news/precorso-di-informatica-1`
- Tutorato (dott.ssa Elena Mariolina Galdi, `elenamarioli.galdi@edu.unife.it`): svolgimento di esercizi, con l'assistenza di un tutor, disponibile anche per chiarimenti

Orari

Lezioni ed esercitazioni: di regola

- Martedì, 10.30 – 13.30: lezioni registrate
- Mercoledì, 10.30 – 13.30: focus group
- Giovedì, 11.00 – 14.00: laboratorio

Focus-group e laboratorio: Aula F8-F9, Chiostro di Santa Maria delle Grazie, Via Fossato di Mortara, 15, oltre a streaming e registrazione
Tutorato:

- L'orario sarà pubblicato alla pagina `http://www.unife.it/scienze/informatica/insegnamenti/programmazione-e-laboratorio/tutorato-didattico`

Ricevimento (via Saragat 1, Blocco A, ufficio 038):

- su appuntamento

Esame

- Prova teorica (max 11 punti, soglia 6): domande e semplici esercizi su tutti gli argomenti del corso (materiale didattico non consentito)
- Prova pratica (max 22 punti, soglia 12): scrittura di un programma in linguaggio C che implementi la specifica assegnata, valutato per identificazione e corretta implementazione di strutture dati e degli algoritmi appropriati alle specifiche, utilizzo efficiente delle risorse, stile (chiarezza, utilizzo di costrutti appropriati, corretta strutturazione)
- E' necessario sostenere le prove nello stesso appello
- L'esame si supera se
  - entrambe le prove sono sufficienti; il voto è pari alla somma dei punteggi (30 se la somma è 30 o 31, 30 e lode se la somma è 32 o 33), oppure
  - una delle prove è insufficiente ma la somma dei punteggi è maggiore o uguale a 18 e si supera una prova orale/pratica integrativa; il voto è uguale alla somma delle prove pratica e teorica

Prove pratiche parziali

P1	P2	P	T
8	10	18	
5	10	15	4
11	4	0	6

E' possibile suddividere la prova pratica in due prove parziali:

- la prima si svolgerà attorno al 10 novembre 2021;
- la seconda all'inizio dell'appello di gennaio 2022.

Ogni prova parziale ha un punteggio massimo di 11 punti. La prova pratica è superata si ottengono almeno 12 punti in totale e almeno 5 in ognuna delle prove parziali; il punteggio è la somma dei due punteggi parziali.
Una volta superata la prova pratica, si può sostenere quella teorica in qualsiasi appello dello stesso anno accademico (fino a settembre 2022); il voto d'esame è determinato come nella slide 10. Se non si supera la prova teorica, occorre ripetere anche quella pratica.

Materiale didattico

- Diapositive usate a lezione, disponibili in Classroom
- Esempi di codice su GitHub
- Manuali degli strumenti
- Testi didattici (disponibili in biblioteca, non obbligatori; vanno benissimo edizioni precedenti):
  - P.Deitel, H.Deitel - Il linguaggio C. Fondamenti e tecniche di programmazione (Ottava edizione) - Pearson
  - A.Bellini, A.Guidi - Linguaggio C (Quinta edizione) - Mc Graw-Hill
- Testi di riferimento:
  - B.W.Kernighan, D.R.Ritchie - Il linguaggio C. Principi di programmazione e manuale di riferimento (Seconda edizione) - Pearson
  - A.Kelley, I.Pohl - C Didattica e Programmazione (Seconda edizione) - Pearson

Diapositive

- Saranno sempre disponibili il lunedì mattina precedente la lezione, in modo che si possano stampare e annotare a lezione
- Saranno via via aggiornate con la versione annotata dal docente
- I titoli degli esempi di codice inseriti nelle diapositive sono link agli esempi pubblicati (il testo del link indica dove trovarli se la diapositiva è stampata). Esempio alla slide 5.

Come si impara a programmare?

Saper programmare richiede

- Capacità di analisi: dato un programma, capire cosa fa (ed eventualmente perché non fa quel che si vuole), per mezzo di un *modello mentale* della macchina, che si costruisce via via con
  - Lezioni
  - Materiale didattico
  - Testi
  - Esercizi: i computer funzionano in modo non intuitivo e a volte ci sorprendono costringendoci ad aggiornare il nostro modello mentale
- Capacità di sintesi: creare un programma che faccia quel che si vuole, combinando le nozioni e le tecniche apprese. Si acquisisce
  - Programmando

Come preparare l'esame

- Nessun prerequisito
- L'esame vale 12 crediti, corrispondenti a 300 ore di cui
  - 96 di attività didattica frontale (lezioni, esercitazioni)
  - 24 di tutorato
  - 180 di studio individuale: da a ora a gennaio 12 settimane effettive, quindi circa 15 ore a settimana, cioè 3 ore al giorno lavorativo.
- Come impiegare le 3 ore? E' necessario svolgere esercizi (al calcolatore e con carta e penna) per imparare a programmare, ma anche per capire la teoria!
  Suggerimento:
  - 1 ora di studio della teoria
  - 2 ore di esercizi
- Cominciando subito: se non si fanno gli esercizi non si capiranno le nozioni di base e sarà difficile seguire gli argomenti avanzati, che saranno trattati più velocemente

Fonti di informazioni

- Pagina ufficiale del corso, con informazioni generali e avvisi: `http://www.unife.it/scienze/informatica/insegnamenti/programmazione-e-laboratorio/`
- Il materiale didattico è distribuito attraverso la piattaforma Google Classroom. Per accedere:
  1. `https://classroom.google.com`
  2. entrare con le proprie credenziali `@edu.unife.it` (le stesse dell'email ricevuta all'immatricolazione).
  3. iscriversi al corso con la password `aebdn3v`
- Esempi di codice, soluzioni di esercizi etc.: `https://github.com/lbrmrc/Programmazione2021/`

Software per esercitarsi

Università degli Studi di Ferrara

Strumenti:

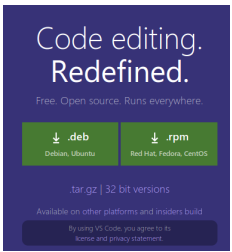
- Editor di testo: Visual Studio Code (https://code.visualstudio.com)
- Compilatore: GCC clang } VISUAL C++
- Debugger: GDB lldb
- Controllo di versione: Git

VIRTUALBOX

E' fornita una macchina virtuale Linux, che permette di avere sul proprio PC un ambiente simile a quello del laboratorio senza cambiare sistema operativo; accessibile alla pagina Classroom. E' l'unico ambiente supportato.

Installazione di Visual Studio Code in laboratorio

Se non lo trovate installato usate la versione *portable*:



1. Andate al sito `http://code.visualstudio.com`
2. scaricate il link `.tar.gz`
3. date il comando `tar -xzvf __file__.tar.gz`, dove `__file__.tar.gz` è il file appena scaricato
4. lanciate il file `code` nella cartella così creata (probabilmente `VSCode-linux-x64`)