



Corso di Informatica A.A. 2009-2010

Introduzione all'informatica

Informazioni pratiche

- Studio N. Itaco: 1G17
- Orario ricevimento:
- Tel. 081-676447
- email: itaco@na.infn.it
- Pagina web del corso:

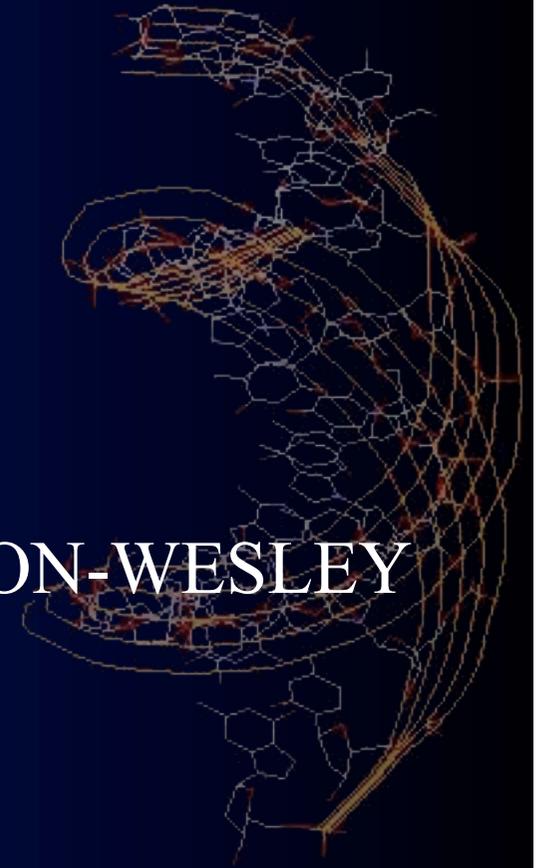
<http://people.na.infn.it/~itaco/informatica/corso.htm>



Testi Consigliati

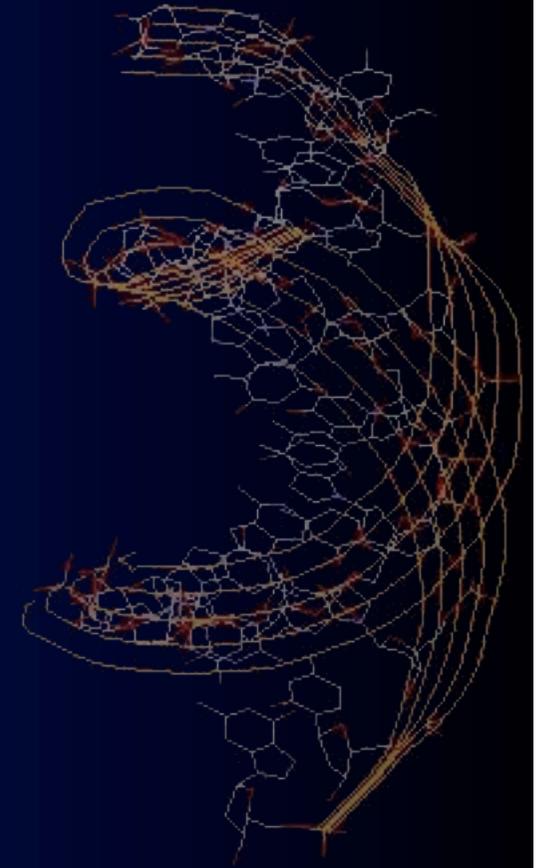
G.M. Schneider, J.L. Gersting
Informatica, APOGEO

Al Kelley, Ira Pohl
C: Didattica e Programmazione, ADDISON-WESLEY



Svolgimento del corso

- Lezioni frontali
- Esercitazioni di laboratorio



Svolgimento del corso

La frequenza delle esercitazioni di laboratorio è obbligatoria

Al termine del corso le prove di laboratorio sono valutate complessivamente secondo il seguente schema

- A → ottimo
- B → buono
- C → sufficiente
- D → insufficiente

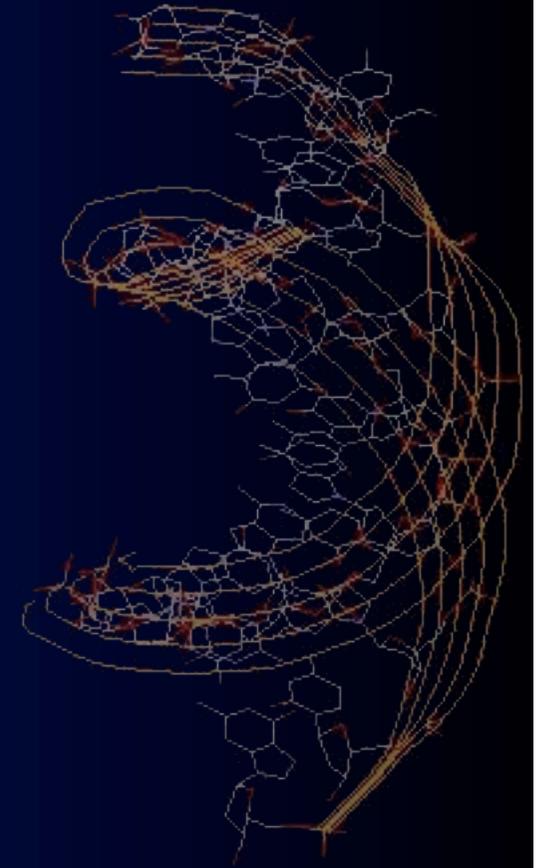
La valutazione contribuisce a determinare il voto finale



Svolgimento del corso

Esame finale

- Prova pratica in laboratorio
- Colloquio orale

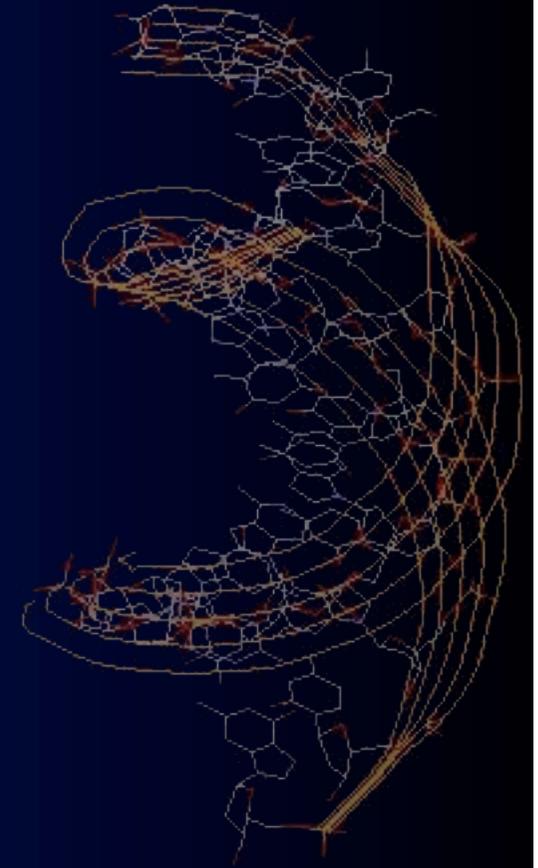


Premessa

Il corso non presuppone conoscenze informatiche precedenti

Corso fatto da Fisici per (futuri) Fisici:

Molto risalto agli aspetti applicativi e pratici



Ma a che serve ...

...l'informatica ad un fisico ?

- Analisi dei dati
- Simulazione di apparati e sistemi fisici complessi
- Controllo di apparecchiature di laboratorio
- Calcoli matematici complessi, sia numerici che simbolici
- Diffusione di risultati e dati rilevanti attraverso le reti
- Consultazione di archivi e ricerche bibliografiche
-

In tutti i settori della fisica teorica e sperimentale: subnucleare, nucleare, della materia, astrofisica, geofisica, biomedica, elettronica...



Obiettivi del corso

Conoscere gli aspetti fondamentali dell'Informatica e della struttura dei calcolatori

Essere in grado di utilizzare un calcolatore in ambiente Unix/Linux

Essere in grado di progettare un algoritmo per risolvere un semplice problema e di implementarlo in un programma in linguaggio C



Cos'è l'informatica ?

L'informatica è lo studio dei computer

Falso!

L'informatica è la scienza della programmazione

Falso!

L'informatica è lo studio degli usi e delle applicazioni dei computer e del software

Falso!

Cos'è l'informatica ?

L'informatica è la scienza della **rappresentazione** e dell'**elaborazione** dell'informazione

L'informatica è lo studio degli **algoritmi**:

- delle loro proprietà formali e matematiche
- delle loro realizzazioni hardware
- delle loro realizzazioni linguistiche
- delle loro applicazioni



Che cos'è un algoritmo ?

Un insieme ordinato di operazioni non ambigue ed effettivamente calcolabili che, quando eseguito, produce un risultato e termina in una quantità di tempo finita.



Esempi di algoritmi

Ricetta di cucina

Sequenza di operazioni per registrare un programma con un videoregistratore

Sequenza di operazioni per montare una tenda

Sequenza di operazioni per moltiplicare due numeri

...

Che cos'è un algoritmo ?

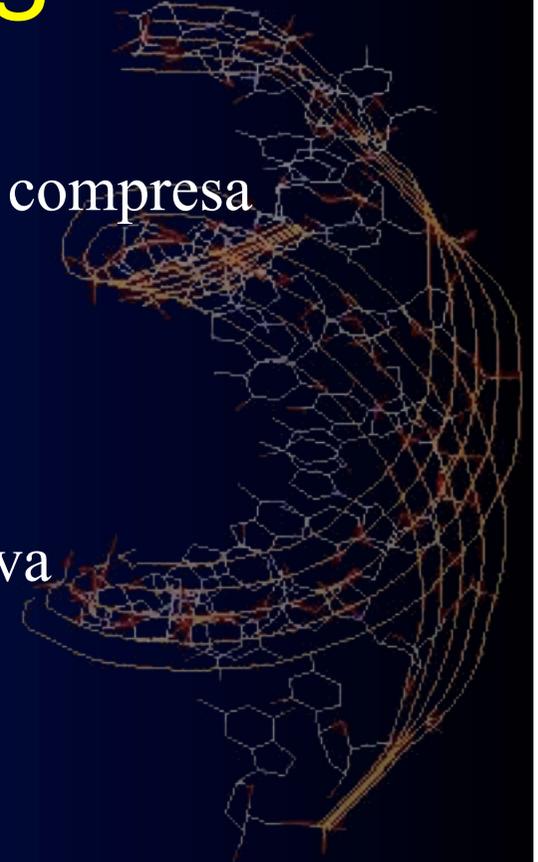
Un insieme ordinato di operazioni non ambigue ed effettivamente calcolabili che, quando eseguito, produce un risultato e termina in una quantità di tempo finita.



...operazioni non ambigue ...

Un'operazione è non ambigua quando può essere compresa dall'agente di calcolo senza necessità di ulteriori semplificazioni o spiegazioni

Operazione non ambigua → operazione primitiva



La ricetta - Torta di mele

1. Prepara la base

Prepara la base

1.1 Prendi una tazza di farina

1.2 Setacciala

1.3 Miscela Farina e 1/2 tazza di burro e 1/4 di tazza d'acqua

2. Prepara il ripieno di

Prepara il ripieno

2.1 Prendi un vasetto da 100 gr di ripieno per torte e versalo in una terrina

2.2 Aggiungi noce moscata, cannella e mescola

3. Versa il ripieno

4. Cuoci in forno a

... effettivamente calcolabili ...

Un'operazione si definisce eseguibile se esiste un processo computazionale che consenta all'agente di calcolo di completare l'operazione con successo

Esempio di operazione non effettivamente calcolabile

- Scrivi il valore decimale esatto di π



Che cos'è un algoritmo ?

Un insieme ordinato di operazioni non ambigue ed effettivamente calcolabili che, quando eseguito, produce un risultato e termina in una quantità di tempo finita.



Un algoritmo ben scritto

La ricetta per cuocere 100 g di pasta:

1. Metti 1l di acqua in una pentola
2. Poni la pentola sul fornello
3. Accendi la fiamma del fornello
4. Finchè l'acqua non bolle ripeti il passo 5.
5. Aspetta un minuto
6. Aggiungi 10 g di sale grosso
7. Leggi sulla confezione della pasta il tempo di cottura
8. Versa la pasta nell'acqua bollente
9. Aspetta il tempo di cui al punto 7.
10. scola la pasta
11. Fine



Scrivere un algoritmo

La ricerca del giusto algoritmo per la soluzione di un dato problema è la parte più creativa del lavoro di un informatico. (Carta e penna)

Ogni algoritmo può essere scomposto in tre tipi fondamentali di operazioni:

- Operazioni sequenziali
- Operazioni condizionali
- Operazioni iterative



Formalizzazione del concetto di algoritmo: perché?

Se siamo in grado di specificare un algoritmo per risolvere un problema, allora possiamo automatizzare la risoluzione del problema

Una volta specificato formalmente un algoritmo possiamo infatti “costruire” un agente di calcolo che esegua le operazioni primitive contenute nell’algoritmo.

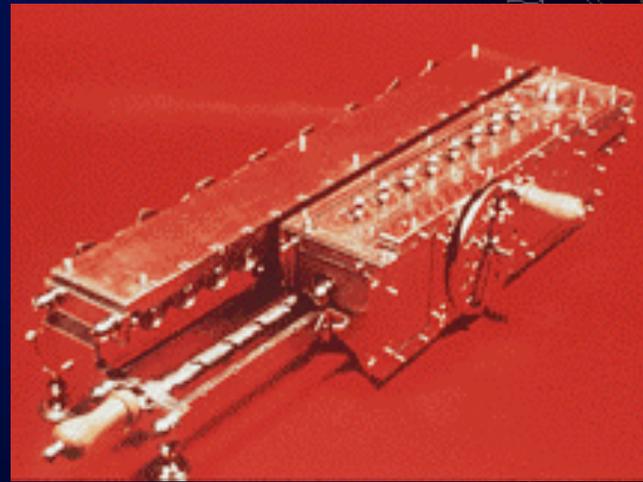
Non tutti i problemi sono risolubili per via algoritmica
(Kurt Gödel)

Automatizzare i calcoli

I primi mezzi automatici di calcolo compaiono nel XVII secolo ad opera di Pascal e Leibniz. Funzionano sulla base di ruote dentate.



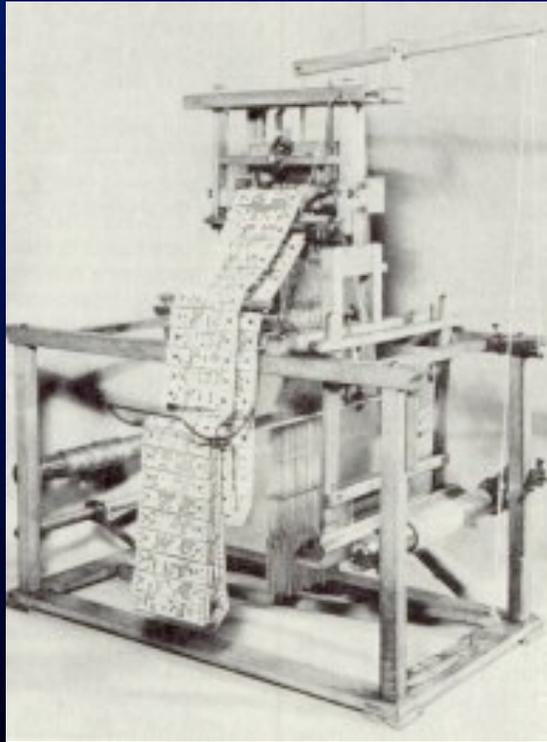
La Pascalina
(Addizione e sottrazione)



La Ruota di Leibniz
(Addizione, sottrazione, moltiplicazione e divisione)

Il primo programma

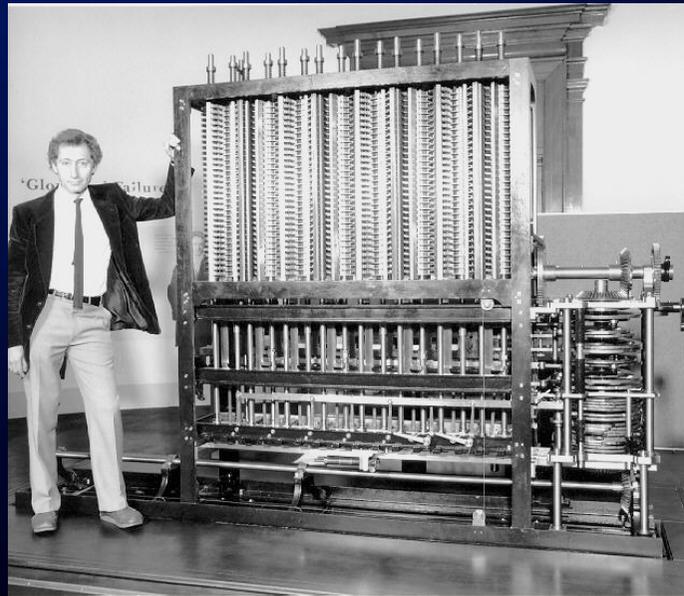
Ma la prima macchina a disporre di un vero programma è il telaio di Jacquard costruito nel 1804 : il disegno della trama viene inserito tramite schede perforate e realizzato automaticamente dalla macchina.



Museo della Seta, S.Leucio (CE)

Il primo calcolatore

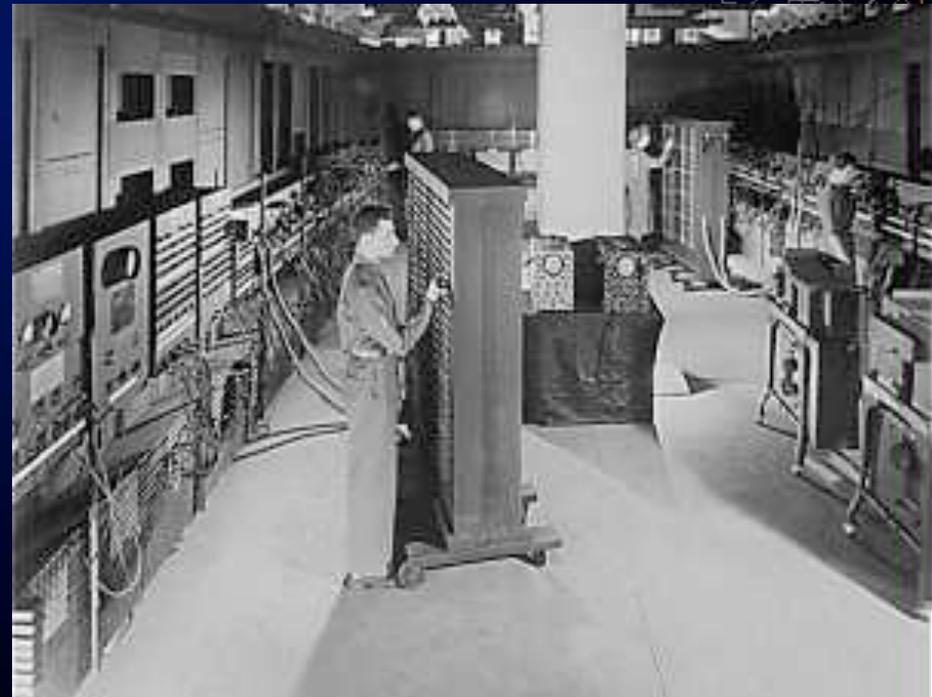
Il primo progetto di un calcolatore vero e proprio, capace di eseguire programmi, si deve a Charles Babbage (1830). L'Analytic engine sfruttava le schede perforate di Jacquard, e una sofisticatissima tecnica meccanica già utilizzata nella precedente Difference engine.



I primi calcolatori elettronici

Motivati principalmente da esigenze belliche i primi grandi calcolatori elettronici a valvole furono realizzati dalle maggiori potenze in conflitto nel corso degli anni 40.

ENIAC: 18000 valvole
30 m lunghezza
30 tonnellate



Il calcolatore moderno

Il principale problema dei calcolatori fino agli anni '40 era la fondamentale differenza fra il modo in cui venivano memorizzati i *dati* e quello in cui venivano realizzati i *programmi*, che non erano memorizzati, ma realizzati esternamente usando connessioni, fili etc.

Fu il genio di Von Neumann che permise di progettare e realizzare, nel 1950, l'EDVAC, il primo *calcolatore a programma memorizzato*. L'EDVAC pesava 8 ton e aveva una memoria di 1024 parole... ma non era nella sostanza diverso dal computer usato per scrivere questa presentazione...

Introduzione alla programmazione in linguaggio C



Breve storia del linguaggio C

Il linguaggio C fu inventato da Dennis Ritchie presso i Bell Telephone Laboratories nel 1972.

Il linguaggio nacque per progettare il sistema operativo UNIX, ma grazie alla sua potenza e versatilità cominciò da subito ad essere ampiamente utilizzato anche al di fuori dei Bell Laboratories.

Perché utilizzare il C ?

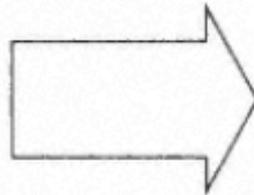
- potente e flessibile
- molto diffuso
- portabile
- possiede poche “parole”
- modulare



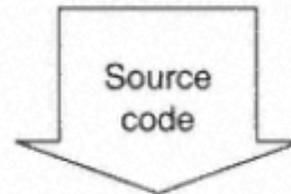
Sviluppo di un programma

1. Scrittura del codice sorgente mediante un editor
2. Compilazione del codice sorgente e conseguente produzione di un file oggetto (obj)
3. Linking per ottenere il codice eseguibile
4. Esecuzione del programma



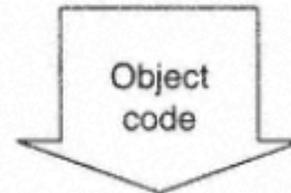


Use editor

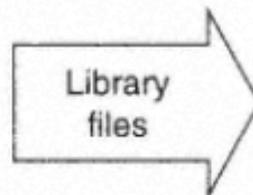


Source code

Compile source file

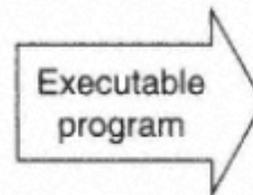


Object code

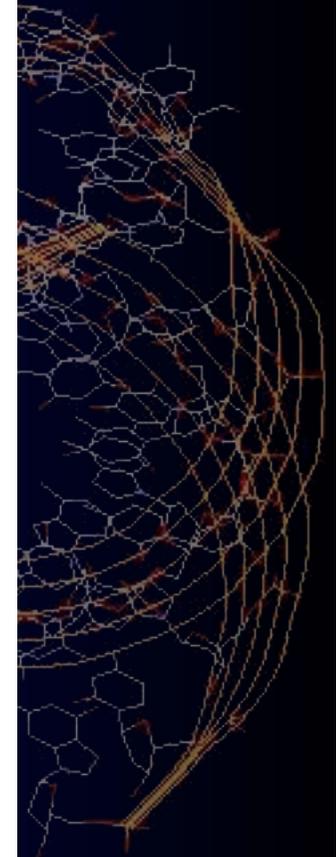


Library files

Link object file



Executable program



Il primo programma in C

```
/* PRIMO - Il primo programma C */
```

commento

```
#include <stdio.h>
```

Header della
libreria

```
main()
```

Funzione
principale

```
{
```

```
    printf("Io so programmare in C!");
```

```
}
```

```
;
```

Ogni istruzione in
C va terminata
con un ;

Funzioni

La linea :

```
printf("Io so programmare in C!");
```

è una chiamata di funzione.

In una chiamata di funzione, il nome della funzione è immediatamente seguito senza spazi da una parentesi aperta. Tra parentesi vi sono i dati da passare alla funzione (argomenti)

- funzioni di libreria
- funzioni user-defined



La funzione `printf`

```
/* STAMPA1: stampa un numero */

#include <stdio.h>

main()
{
    printf("\n Stampa un numero intero:\n\n");    /*stampa stringa <1>*/
    printf("%d", 1234);                          /*stampa numero <2>*/
}

/* Note:
<1> Se la stringa passata alla printf non contiene il carattere di
percento %, viene stampata tale e quale.
<2> Se la stringa contiene il carattere di percento %, questo indica
l'inizio di una specifica di stampa. Ad esempio, %d indica di
stampare un numero intero, che viene passato come secondo parametro.
Dopo la stampa del numero, il cursore non torna a capo.
*/
```

Specifica di
stampa

Funzione printf

`printf` e' una funzione della libreria `stdio` che permette ad un programma di passare informazioni allo standard output, il cui dispositivo associato e' tipicamente il terminale grafico. Il formato con cui sono presentati i numeri e le stringhe di caratteri e' definito da un insieme di regole e simboli di formattazione.

```
printf(formato, arg1, arg2...);
```

`formato` : stringa di caratteri che contiene le informazioni per la formattazione, i descrittori del formato

`arg1, arg2...` : nomi di variabili di cui si vuole stampare il contenuto

```
/* Program to calculate the product of two numbers. */
#include <stdio.h>

int a,b,c;

int product(int x, int y);

int main()
{
    /* Get the first number */
    printf("Enter a number between 1 and 100: ");
    scanf("%d", &a);

    /* Get the second number */
    printf("Enter another number between 1 and 100: ");
    scanf("%d", &b);

    /* Calculate and display the product */
    c = product(a, b);
    printf ("%d times %d = %d\n", a, b, c);

    return 0;
}

/* Function returns the product of the two values provided */
int product(int x, int y)
{
    return (x * y);
}
```