ROMA TRE
UNIVERSITÀ DEGLI STUDI

Dipartimento di Ingegneria
Industriale, Elettronica e Meccanica

Collegio Didattico di Ingegneria Meccanica

Elementi di Informatica

Variabili e istruzioni

Giordano Da Lozzo e Giuseppe Sansonetti

L'informatica permette di risolvere problemi

hai un problema
che vuoi risolvere

trovi una procedura che ti
permette di risolvere il problema
(su ogni possibile insieme di dati)

algoritmo

trasformi la procedura in una
sequenza di istruzioni che possono
essere eseguite dal calcolatore

programma

se vuoi risolvere
il problema su un certo
insieme di dati

istanza

rappresenti i dati nel calcolatore e fai eseguire
al calcolatore la sequenza di istruzioni che
permettono di risolvere il problema

esecuzione del programma

Scopo del giorno

Introdurre concetti di base del linguaggio C ed incominciare ad usarli.

Iniziate a giocare con la programmazione!

Nota: alcune parti dei programmi che scriveremo rimarranno oscure ancora per qualche tempo.

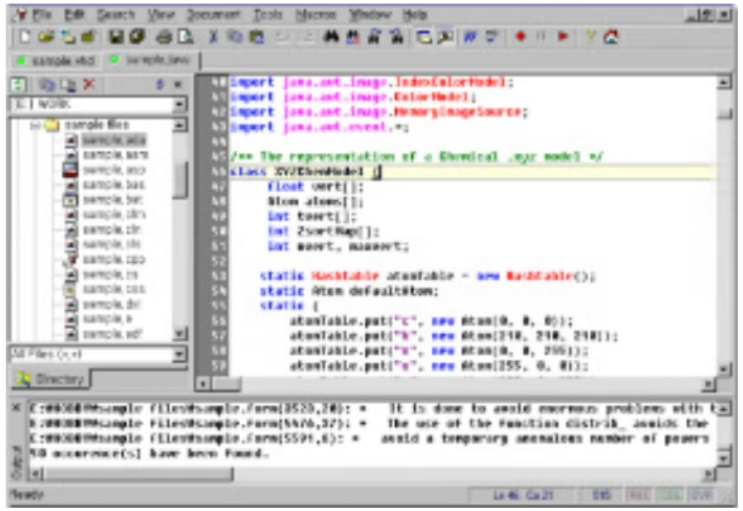
Attori nel gioco dei programmi

Il programmatore → colui che progetta e realizza il programma.

L'utente → colui che utilizza il programma.

I programmi commerciali vengono utilizzati attraverso una interfaccia utente che fornisce all'utente una maniera intuitiva (interazione tramite tastiera, mouse, schermo, altoparlanti) per utilizzare le funzionalità messe a disposizione dal programma.

quello che vede l'utente



quello che scrive il programmatore

Sintassi e Semantica

La sintassi si occupa delle regole grammaticali che permettono di scrivere frasi corrette e "ben formate" del linguaggio C.

Quest'istruzione si scrive così…

La semantica si occupa del significato delle frasi ben fornate, ovvero dell' "interpretazione" del linguaggio.

Quest'istruzione ha come effetto…

Istruzioni

Programma

È una sequenza di istruzioni.

Quando un programma viene eseguito, vengono eseguite tutte le sue istruzioni

⇒ Una alla volta!

⇒ Dall'alto verso il basso! (con alcune eccezioni ...)

Istruzione di stampa

E’ possibile scrivere un'istruzione di stampa!

⇒ La semantica di un'istruzione di stampa è che viene stampato "qualcosa" (una frase, un valore...) sullo schermo dell’utente

qua dentro ci va quello che vuoi stampare fra doppi apici " "

⇒ La sintassi è printf( );

```
printf("STATE IMPARANDO UN SACCO DE ROBBA!");
```

Quello che scrive il programmatore

Quello che vede l'utente che esegue il programma

Utilizzo di printf

Per poter utilizzare printf, così come altre funzioni di input e output, è necessario avvertire il pre-compilatore C di includere nel nostro programma i riferimenti alla libreria standard di input/output.

Una libreria è una raccolta di programmi che hanno uno scopo simile.

All'inizio del programma, scrivi la direttiva

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

Spazi e ritorni a capo

Lo spazio è un carattere come tutti gli altri e come tale va trattato!

```
printf("STATE");
printf("IMPARANDO");
printf("UN");
printf("SACCO");
printf("DE");
printf("ROBBA!");
```

Le istruzioni printf successive alla prima scrivono a partire dalla posizione immediatamente successiva a quella dell'ultimo carattere scritto dall'istruzione printf precedente.

Per andare a capo, si utilizza il carattere newline \n

```
printf("STATE IMPARANDO \nUN SACCO DE ROBBA!");
```

Struttura del programma

MAIN() o Main() non vanno bene!

La funzione main() denota l'inizio del programma

Inclusione libreria

Questa linea è un commento: Illustra, ma non modifica, il significato del programma

```
1 #include <stdio.h>
2
3 /* Programma un sacco de robba */
4 int main(){
5     printf("STATE IMPARANDO UN SACCO DE ROBBA!");
6 }
```

Il significato di int e delle parentesi tonde sarà discusso in futuro

Il corpo del programma (ovvero l'insieme di istruzioni che devono essere eseguite quando il programma viene avviato) è racchiuso tra parentesi graffe

{ corpo }

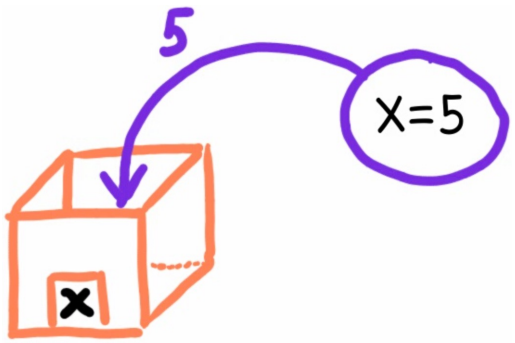
Variabili

Variabile

In matematica, una variabile è un carattere alfabetico che rappresenta un numero arbitrario, sconosciuto, o non completamente specificato.

In informatica, una variabile è una porzione di memoria destinata a contenere dei dati che potranno essere acceduti o modificati durante l'esecuzione di un programma.

- *In maniera più semplice:* in informatica, una variabile è un contenitore di valori.



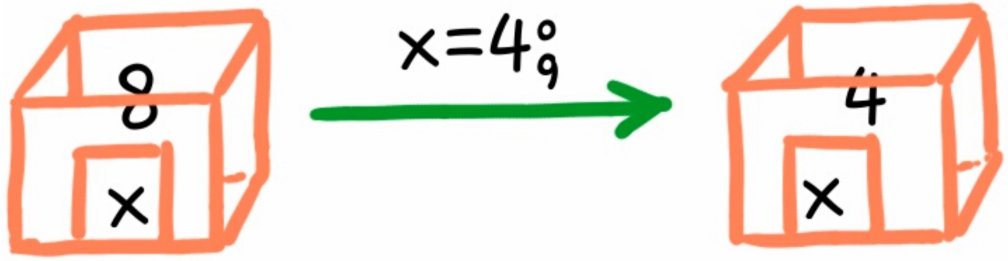
Che ci si fa con una variabile?

- Ci si memorizza un valore dentro (accesso in scrittura)
  - ➢ metti un valore nel contenitore

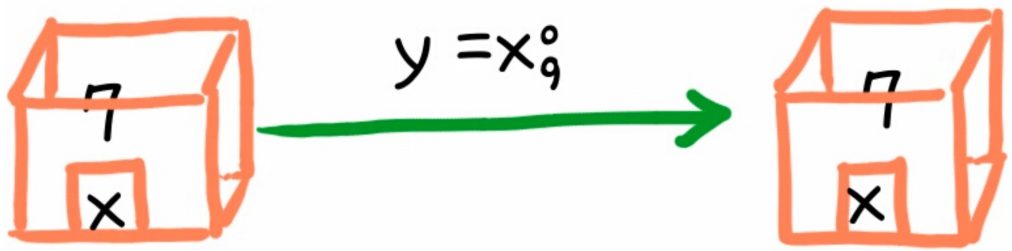
- Si accede al valore della variabile (accesso in lettura)
  - ➢ utilizza il valore nel contenitore

Note sulla scrittura e lettura di variabili

Se un programma scrive dentro ad una variabile che già conteneva un valore, il valore precedentemente contenuto viene perso e la variabile conterrà il nuovo valore.



Se un programma legge il valore contenuto dentro ad una variabile, quello stesso valore è ancora memorizzato dentro la variabile dopo la lettura.



Nome o identificatore

Le variabili hanno un nome che permette di identificarle.

- Ogni variabile deve avere un nome diverso.
- Il nome dovrebbe essere esplicativo dello scopo della variabile
  - Es: *discriminante* è un nome migliore di *ax23Dd* per una variabile che è destinata a memorizzare il discriminante di un'equazione di 2° grado.
- Un nome è una sequenza di caratteri alfanumerici, il cui primo carattere deve essere alfabetico (oppure un underscore _) e in genere è preferibilmente alfabetico minuscolo. Ogni parola che costituisce il nome, dopo la prima, dovrebbe iniziare con una maiuscola.
  - Es: *radiceReale1, primoNumeroDellaSequenza, minimoCorrente, sequenza3Numeri*

Tipo

Le variabili hanno un tipo che rappresenta l'insieme dei valori che la variabile può assumere e l'insieme di operazioni permesse su quei valori.

Durante l'esecuzione di un programma, una variabile può assumere valori diversi, ma sempre dello stesso tipo.

- Es: una variabile che in un certo momento memorizza un intero non può memorizzare un carattere dell'alfabeto in un altro momento dell'esecuzione di un programma.

Alcuni tipi

int ⇒ insieme di interi

float ⇒ insieme di reali in virgola mobile

char ⇒ insieme di caratteri

Dichiarazione

Per poter essere utilizzate, le variabili vanno dichiarate (operazione che corrisponde a dire “utilizzerò una variabile x di tipo int" oppure “utilizzerò una variabile $numeroReale$ di tipo float").

Sintassi: tipo nome;

Esempi: int x; float radice; int y, numero, resto;

Semantica: la dichiarazione serve a dire al calcolatore "utilizzerò una variabile nome di tipo tipo, quindi prepara un contenitore (un’area di memoria) per valori di tipo tipo con nome nome”

Dichiarazione

La dichiarazione di una variabile comporta la "allocazione" in memoria centrale di uno spazio, destinato a memorizzare i valori assunti dalla variabile durante l'esecuzione del programma.

Il tipo di una variabile determina quanto spazio viene allocato in memoria per la variabile stessa.

Le dichiarazioni di variabile sono (in genere) all'inizio del corpo del programma.

Assegnazione o assegnamento

Dopo aver dichiarato una variabile, si può memorizzare un valore nella variabile.

una porzione di codice che ha un «tipo»

Sintassi: *nome* = *espressione*; ⟶ occhio al punto e virgola

operatore di assegnazione

variabile nella quale vogliamo scrivere

Esempi: $x = 5;$ $y = (1.5 + 2)/4;$

Semantica: il valore dell'espressione espressione a destra dell'operatore di assegnazione = viene calcolato e memorizzato nella variabile nome a sinistra dell'operatore di assegnazione.

Assegnazione o assegnamento

L'assegnazione di un valore ad una variabile ha come effetto la memorizzazione del valore nella variabile.

Dichiarazione con inizializzazione

Inizializzazione: prima assegnazione di un valore ad una variabile.

Dichiarazione con inizializzazione: dichiara ed inizializza una variabile nella stessa istruzione.

Sintassi: $tipo\ nome\ =\ espressione;$

Esempio: $int\ x = 5;$

Semantica: combina la semantica di dichiarazione ed assegnazione.

Accesso

Operazione che permette di accedere al valore memorizzato all'interno di una variabile.

Sintassi: *nome*

Ovvero l'accesso ad un valore memorizzato da una variabile si fa scrivendo il nome della variabile.

Regola:

Se *nome* compare a sinistra di un operatore di assegnazione la variabile viene usata per memorizzare un valore. Altrimenti, la variabile viene usata per accedere al valore che memorizza.

Semantica: utilizza il valore memorizzato in *nome* al posto di *nome* nella porzione di codice in cui la variabile compare.

Esempi

$x = y + z;$ /*accede ad y e z per sommarne i valori
e memorizzare il risultato dentro ad x */

$x = x + 2;$ /* accede al valore di x, somma 2 a tale valore e
memorizza il risultato nella (stessa) variabile x */

Istruzioni di input/output

Stampa

La funzione printf permette di stampare una sequenza di caratteri, ovvero una stringa, che deve essere specificata fra doppi apici "..."

```
printf("STATE IMPARANDO UN SACCO DE ROBBA!");
```

La funzione printf permette anche di stampare dei valori, ad esempio di variabili ed in generale di espressioni, nella seguente maniera:

- La stringa deve indicare, per ciascun valore da stampare, il formato di stampa di quel valore, preceduto dal simbolo di percentuale %
- I valori da stampare devono essere riportati fra parentesi tonde

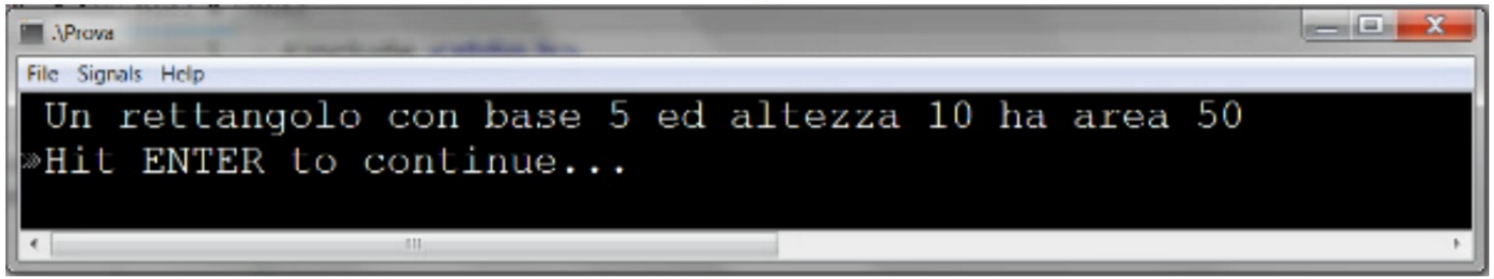
Esempi di stampe

```c
#include <stdio.h>
int main(){
    int x = 5;
    printf("La variabile x vale %d", x);
}
```

d è il formato di stampa per valori di tipo int

La stringa e ciascun valore sono separati da virgole. Per ottenere il valore memorizzato nella variabile x si scrive semplicemente x (accesso in lettura).

```c
#include <stdio.h>
int main(){
    int x = 5;
    int y = 10;
    printf("Un rettangolo con base %d ed altezza %d ha area %d", x, y, x*y);
}
```



I valori non ne devono essere necessariamente variabili, ma espressioni del tipo del formato

Lettura

- La funzione scanf permette di leggere valori immessi dall'utente.

Sintassi: scanf ("%formato", &variabile);

- %formato indica il formato dei valori da leggere
  - (Ad es. %d per leggere un intero) come per *printf*
  - &variabile indica l'indirizzo della memoria al quale si trova la variabile.

Semantica: il programma si pone in attesa che l'utente introduca un valore con formato *d*. Quando l'utente introduce un valore, tale valore viene memorizzato nella variabile il cui indirizzo è *&variabile*.

Analogamente ad una istruzione di assegnazione, una lettura ha come effetto quello di memorizzare un valore in una variabile.

Per comunicare all'utente che deve immettere dei valori, una istruzione di lettura è in genere preceduta da una istruzione di stampa.

```c
# include <stdio.h>

/* programma che calcola l'area di un rettangolo */
int main() {
    int base, altezza, area; // variabili per le misure

    /* INPUT */
    printf("Ciao utente, introduci base ed altezza del rettangolo.\n");
    scanf("%d", &base);
    scanf("%d", &altezza);

    /* calcola l'area e visualizza il risultato */
    area = base * altezza;
    printf("\nL'area del rettangolo e' %d.\n", area);
}
```

Altre risorse

- Bellini, Guidi: Linguaggio C — Capitoli 4.4, 5.1—5.4